

UNIVERSIDADE FEDERAL DE ALAGOAS
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA CIVIL
ESTRUTURAS

CHRISTIANO AUGUSTO FERRARIO VÁRADY FILHO

**UM AMBIENTE COMPUTACIONAL DE ALTO DESEMPENHO PARA
CÁLCULO DE DESLOCAMENTOS USANDO CORRELAÇÃO DE
IMAGENS DIGITAIS**

MACEIÓ/AL

2016

CHRISTIANO AUGUSTO FERRARIO VÁRADY FILHO

**UM AMBIENTE COMPUTACIONAL DE ALTO
DESEMPENHO PARA CÁLCULO DE DESLOCAMENTOS
USANDO CORRELAÇÃO DE IMAGENS DIGITAIS**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil da Universidade Federal de Alagoas como requisito para obtenção do título de Mestre em Engenharia Civil na área de concentração em Estruturas.

Orientador: Prof. Dr. William Wagner Matos Lira

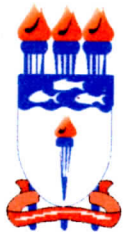
MACEIÓ/AL
2016

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecária Responsável: Helena Cristina Pimentel do Vale

- V287u Várady Filho, Christiano Augusto Ferrario.
Um ambiente computacional de alto desempenho para cálculo de deslocamentos usando correlação de imagens digitais / Christiano Augusto Ferrario Várady Filho. – 2016.
116 f. : il., graf.
- Orientador: William Wagner Matos Lira.
Dissertação (Mestrado em Engenharia Civil) – Universidade Federal de Alagoas. Centro de Tecnologia. Programa de Pós-Graduação em Engenharia Civil. Maceió, 2016.
- Bibliografia: f. 110-113.
1. Engenharia civil. 2. Computação de alto desempenho. 3. Correlação de imagens digitais. 4. Processamento de imagens. I. Título.

CDU: 624:004



**UM AMBIENTE COMPUTACIONAL DE ALTO DESEMPENHO PARA O CÁLCULO DE
DESLOCAMENTO USANDO CORRELAÇÃO DE IMAGENS DIGITAIS**

CHRISTIANO AUGUSTO FERRARIO VÁRADY FILHO

Dissertação submetida à banca examinadora do Programa de Pós-Graduação em Engenharia Civil da Universidade Federal de Alagoas e aprovada no dia 04 do mês de abril do ano de 2016.

Banca Examinadora:

Prof. Dr. William Wagner Matos Lira
Orientador – PPGEC/UFAL

Prof. Dr. Eduardo Nobre Lages
(Avaliador Interno - PPGEC/UFAL)

Prof. Dr. Adeildo Soares Ramos Junior
(Avaliador Interno - PPGEC/UFAL)

Prof. Dr. Rodrigo Nogueira de Codes
(Avaliador Externo – UFERSA)

AGRADECIMENTOS

A meus pais, por me ensinarem a buscar meus objetivos.

A Paula, por participar comigo de todos os momentos dessa vitória.

Ao professor William Lira, pela orientação em todas as pesquisas.

Ao professor Eduardo Toledo, pelo auxílio nas muitas solicitações que fiz ao PPGEC.

Aos colegas do LCCV Ricardo e Catarina Fernandes, Lucas Gouveia e Tiago Lobo pelo apoio no desenvolvimento do trabalho.

Ao colega Romildo Escarpini, pelas muitas discussões e pelo apoio durante toda a dissertação.

Ao CNPq, pela bolsa dispensada.

“The mind itself has no needs, except for those it creates itself. Is undisturbed, except for its own disturbances. Knows no obstructions, except those from within”

Marcus Aurelius - Meditations

RESUMO

O presente trabalho apresenta um ambiente computacional de alto desempenho que utiliza técnicas de correlação de imagens digitais para determinação de grandezas físicas associadas a problemas de engenharia. Sua arquitetura computacional integra diversas tecnologias avançadas para o cálculo dos campos de deslocamentos de elementos estruturais a partir de ensaios. A metodologia utilizada prevê o estudo de conceitos, formulações e técnicas de arquitetura de *software*, processamento de imagens, correlação de imagens digitais e computação de alto desempenho, além de incorporar procedimentos específicos em um ambiente único para a determinação de grandezas físicas. Dentre os principais procedimentos utilizados na arquitetura apresentada, pode-se citar as abordagens locais e globais de correlação de imagens digitais, métodos de otimização não-linear e interpolações bidimensionais. Também são incorporadas ao ambiente computacional estratégias de computação de alto desempenho para alcançar ganhos de performance na determinação dos campos de deslocamentos e deformações usando a correlação de imagens digitais. Uma comparação com o método *Scale Invariant Feature Transform* e com o método Q4-DIC de análise de imagens também são realizadas. Um protótipo computacional é desenvolvido com o objetivo de validar o ambiente de alto desempenho apresentado, permitindo o monitoramento de elementos estruturais através da correlação de imagens digitais. Também são realizados estudos de casos que permitem a verificação de tecnologias incorporadas ao protótipo apresentado, incluindo análises quantitativas de resultados e medição de tempo computacional.

Palavras-chaves: Correlação de imagens digitais. Processamento de imagens. Computação de alto desempenho.

ABSTRACT

This work proposes a high performance computing environment using digital image correlation techniques to determine physical quantities associated with engineering problems. Software architecture supports this computing environment, integrating several advanced technologies for calculation of displacement and strain fields of structural elements from testing. The methodology applies the study of concepts, formulations and techniques for image processing, digital image correlation, high performance computing and software architecture. The methodology also includes specific procedures in a single environment for the evaluation of physical quantities. Among the main procedures used in the presented software architecture, one can cite the digital image correlation techniques known as Full-Field and Subset, non-linear optimization methods and two-dimensional interpolations. In addition, high performance computing strategies are included into the computing environment to achieve performance speed-ups on evaluating the displacement fields using digital image correlation. Comparisons with *Scale Invariant Feature Transform* and Q4-DIC are also evaluated. Following, the development of a computer prototype has the purpose of validating the presented high performance environment, allowing the calculation of physical quantities in structural elements through the correlation of digital images. Then, submission of case studies into the prototype validates data acquired from technologies built into the prototype, including quantitative analysis of results and measurement of computational time.

Keywords: Digital image correlation. Image processing. High performance computing

LISTA DE FIGURAS

Figura 1 – Relação entre áreas da computação gráfica.	13
Figura 2 – Fluxograma da Pesquisa.	20
Figura 3 – Domínio espacial.	23
Figura 4 – Domínio de intensidades.	23
Figura 5 – Representação da correlação de imagens digitais.	24
Figura 6 – Grade de células para análise seguindo o método <i>Subset</i>	27
Figura 7 – Grade de pontos para análise seguindo o método <i>Full-Field</i>	28
Figura 8 – Função do campo de deslocamentos.	29
Figura 9 – Subcélula utilizada em interpolação 3x3.	34
Figura 10 – Representação de processamento paralelo.	36
Figura 11 – Pseudo-algoritmo para SIFT.	37
Figura 12 – Representação multiescalar.	37
Figura 13 – Diagrama do modelo geral do ambiente.	39
Figura 14 – Diagrama do módulo de pré-processamento.	40
Figura 15 – Diagrama do módulo de análise por CID.	41
Figura 16 – Diagrama do módulo de pós-processamento.	42
Figura 17 – Estrutura interna do módulo de análise.	43
Figura 18 – Fases do processo de cálculo do coeficiente de correlação.	44
Figura 19 – Diagrama simplificado do cálculo do coeficiente de correlação.	45
Figura 20 – Fases do processo de otimização não-linear.	46
Figura 21 – Diagrama simplificado com processo de otimização não-linear - abordagem local de análise.	46
Figura 22 – Conversão entre imagem e função contínua.	47
Figura 23 – Fases do processo de otimização não-linear.	47
Figura 24 – Diagrama simplificado do processo de interpolação.	48
Figura 25 – Fases do processo de computação de alto desempenho.	49
Figura 26 – Diagrama simplificado para Computação de Alto Desempenho.	50
Figura 27 – Adaptações realizadas ao fluxo de Ferraz (2013).	51
Figura 28 – Pseudo-Algoritmo do Método de Levenberg-Marquardt.	57
Figura 29 – Imagens original e deformada para o exemplo 1.	60
Figura 30 – Erro absoluto na direção X – Exemplo 1.	61
Figura 31 – Erro absoluto na direção Y – Exemplo 1.	61
Figura 32 – Número de iterações por tamanho de célula – Exemplo 1.	62
Figura 33 – Tempo de processamento por tamanho de célula - Exemplo 1.	62
Figura 34 – Coeficiente de correlação por tamanho de célula - Exemplo 1.	63
Figura 35 – Imagens original e deformada para o exemplo 2.	64

Figura 36 – Erro absoluto na direção X – Exemplo 2.	65
Figura 37 – Erro absoluto na direção Y – Exemplo 2.	65
Figura 38 – Número de iterações por tamanho de célula – Exemplo 2.	66
Figura 39 – Tempo de processamento por tamanho de célula - Exemplo 2.	66
Figura 40 – Coeficiente de correlação por tamanho de célula - Exemplo 2.	67
Figura 41 – Imagens original e deformada para o exemplo 3.	67
Figura 42 – Erro absoluto na direção X – Exemplo 3.	68
Figura 43 – Erro absoluto na direção Y – Exemplo 3.	69
Figura 44 – Número de iterações por tamanho de célula – Exemplo 3.	70
Figura 45 – Tempo de processamento por tamanho de célula - Exemplo 3.	70
Figura 46 – Coeficiente de correlação por tamanho de célula - Exemplo 3.	71
Figura 47 – Imagem original para os exemplos 4 e 5.	72
Figura 48 – Imagem deformada para o exemplo 4.	72
Figura 49 – Erro absoluto na direção X – Exemplo 4.	73
Figura 50 – Erro absoluto na direção Y – Exemplo 4.	74
Figura 51 – Número de iterações por tamanho de célula – Exemplo 4.	75
Figura 52 – Tempo de processamento por tamanho de célula - Exemplo 4.	75
Figura 53 – Coeficiente de correlação por tamanho de célula - Exemplo 4.	76
Figura 54 – Imagem deformada para o exemplo 5. Fonte: Autor.	77
Figura 55 – Erro absoluto na direção X – Exemplo 5.	78
Figura 56 – Erro absoluto na direção Y – Exemplo 5.	79
Figura 57 – Número de iterações por tamanho de célula – Exemplo 5.	80
Figura 58 – Tempo de processamento por tamanho de célula - Exemplo 5.	80
Figura 59 – Coeficiente de correlação por tamanho de célula - Exemplo 5.	81
Figura 60 – Escopo da análise realizada na Seção 5.	81
Figura 61 – Grade de células para avaliação de desempenho.	82
Figura 62 – Tempo computacional para o exemplo 1.	84
Figura 63 – Escalabilidade para o exemplo 1.	84
Figura 64 – <i>Speed-up</i> para o exemplo 1.	85
Figura 65 – Tempo computacional para o exemplo 2.	86
Figura 66 – Escalabilidade para o exemplo 2.	86
Figura 67 – <i>Speed-up</i> para o exemplo 2.	87
Figura 68 – Tempo computacional para o exemplo 3.	88
Figura 69 – Escalabilidade para o exemplo 3.	88
Figura 70 – <i>Speed-up</i> para o exemplo 3.	89
Figura 71 – Tempo computacional para o exemplo 4.	90
Figura 72 – Escalabilidade para o exemplo 4.	90
Figura 73 – <i>Speed-up</i> para o exemplo 4.	91
Figura 74 – Tempo computacional para o exemplo 5.	91
Figura 75 – Escalabilidade para o exemplo 5.	92

Figura 76 – <i>Speed-up</i> para o exemplo 5.	92
Figura 77 – Pontos detectados pelo método SIFT - Exemplo 1.	94
Figura 78 – Erros absolutos entre SIFT e DIC para o eixo X - Exemplo 1.	95
Figura 79 – Erros absolutos entre SIFT e DIC para o eixo Y - Exemplo 1.	95
Figura 80 – Pontos detectados pelo método SIFT - Exemplo 2.	96
Figura 81 – Erros absolutos entre SIFT e DIC para o eixo X - Exemplo 2.	96
Figura 82 – Erros absolutos entre SIFT e DIC para o eixo Y - Exemplo 2.	97
Figura 83 – Pontos detectados pelo método SIFT - Exemplo 3.	97
Figura 84 – Erros absolutos entre SIFT e DIC para o eixo X - Exemplo 3.	98
Figura 85 – Erros absolutos entre SIFT e DIC para o eixo Y - Exemplo 3.	98
Figura 86 – Pontos detectados pelo método SIFT - Exemplo 4.	99
Figura 87 – Erros absolutos entre SIFT e DIC para o eixo X - Exemplo 4.	99
Figura 88 – Erros absolutos entre SIFT e DIC para o eixo Y - Exemplo 4.	100
Figura 89 – Pontos detectados pelo método SIFT - Exemplo 5.	101
Figura 90 – Erros absolutos entre SIFT e DIC para o eixo X - Exemplo 5.	101
Figura 91 – Erros absolutos entre SIFT e DIC para o eixo Y - Exemplo 5.	102
Figura 92 – Imagem real do ensaio de compressão diametral.	103
Figura 93 – Erro entre Q4-DIC e protótipo - Eixo X.	104
Figura 94 – Erro entre Q4-DIC e protótipo - Eixo Y.	104

LISTA DE TABELAS

Tabela 1 – Dados para o exemplo 1.	60
Tabela 2 – Resultados dos parâmetros para o Exemplo 1.	60
Tabela 3 – Dados para o exemplo 2.	64
Tabela 4 – Resultados dos parâmetros para o Exemplo 2.	64
Tabela 5 – Dados para o exemplo 3.	68
Tabela 6 – Resultados dos parâmetros para o Exemplo 3.	68
Tabela 7 – Dados para o exemplo 4.	73
Tabela 8 – Resultados dos parâmetros para o Exemplo 4.	73
Tabela 9 – Dados para o exemplo 5.	77
Tabela 10 – Resultados dos parâmetros para o Exemplo 5.	78
Tabela 11 – Valores para os erros entre o protótipo e CorreliQ4.	103

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Revisão de Literatura	14
1.2	Justificativa	19
1.3	Objetivos e Metas	19
1.4	Metodologia	20
1.5	Delimitação do Trabalho	21
1.6	Escopo do Trabalho	22
2	CONCEITOS BÁSICOS	23
2.1	Análise de Imagens Digitais	23
2.2	Correlação de Imagens Digitais	24
2.3	Abordagem Global/Local de Análise	26
2.4	Funções de Mapeamento/Modelo de Transformação	28
2.5	Otimização Não-Linear	31
2.6	Interpolação	33
2.7	Computação de Alto Desempenho	35
2.8	Método SIFT	36
3	AMBIENTE COMPUTACIONAL DE ALTO DESEMPENHO	39
3.1	Estrutura do Ambiente	39
3.2	Componentes do Módulo de Análise	42
3.2.1	Coeficiente de Correlação	43
3.2.2	Otimização Não-Linear	45
3.2.3	Interpolação	47
3.3	Computação de Alto Desempenho	49
4	PROTÓTIPO DO AMBIENTE	51
4.1	Tipo de Abordagem e Campo de Deslocamentos	52
4.2	Coeficiente de Correlação	52
4.3	Interpolação	53
4.4	Otimização Não-Linear: Métodos de Newton-Raphson e Levenberg-Marquardt	54
4.4.1	Estimativa Inicial	54
4.4.2	Refinamento da Solução	55
5	RESULTADOS	58
5.1	Geração das Imagens	59

5.2	Validação do Protótipo	59
5.2.1	Exemplo 1	59
5.2.2	Exemplo 2	63
5.2.3	Exemplo 3	67
5.2.4	Exemplo 4	71
5.2.5	Exemplo 5	76
5.3	Computação de Alto Desempenho	81
5.3.1	Exemplo 1	83
5.3.2	Exemplo 2	85
5.3.3	Exemplo 3	87
5.3.4	Exemplo 4	89
5.3.5	Exemplo 5	91
5.4	Método SIFT	93
5.4.1	Procedimento de Análise	93
5.4.2	Resultados	94
5.5	Q4-DIC	102
5.5.1	Procedimento de Análise	102
5.5.2	Resultados	103
6	CONCLUSÃO	105
6.1	Trabalhos Futuros	106
	REFERÊNCIAS	110

1 INTRODUÇÃO

Em diversas situações de Engenharia, existe a necessidade de monitoramento de elementos estruturais que estejam em condições desfavoráveis, de maneira a garantir que o dimensionamento deles seja capaz de suportar as solicitações atuantes. Atualmente, o acompanhamento *in loco* com equipamentos de última geração permite uma análise bastante minuciosa das deformações sofridas, assim como estimar a vida útil do elemento.

O monitoramento realizado em ensaios durante o desenvolvimento de materiais estruturais inovadores calcula os parâmetros que definem o comportamento do material. Dessa forma, a aplicação do material em larga escala depende da definição completa de suas características mecânicas. Normalmente, tais ensaios são realizados com elementos em escala reduzida para diminuição de custos.

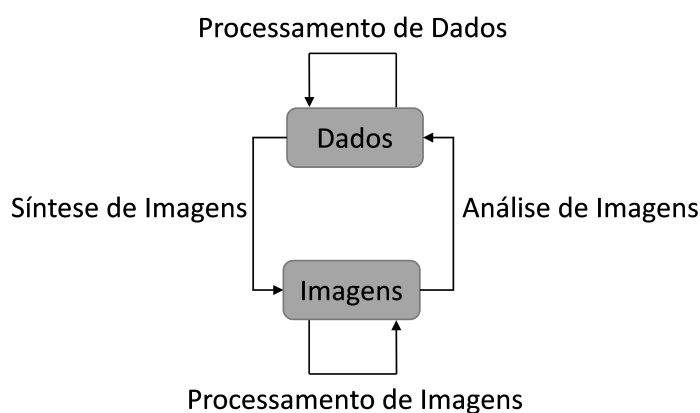
Dependendo da escala utilizada, o aferidor pode encontrar várias dificuldades para inserir equipamentos de medição sem afetar o comportamento geral do objeto de ensaio pela incompatibilidade entre o tamanho dos instrumentos e as dimensões do modelo estudado.

Dentre alguns equipamentos utilizados para medição de deformações, citam-se os sensores elétricos (*Strain Gauges*) ou sensores de cordas vibrantes. Para informações mais específicas sobre sensores de monitoramento estrutural, recomenda-se Assis (2007). Tais sensores são inseridos no objeto de estudo em locais específicos, onde se deseja medir a deformação ocorrida. Esse tipo de instrumento gera informações de deformação para a direção onde eles são inseridos. A avaliação de campos complexos de deslocamentos envolve a inserção de vários desses sensores em diferentes direções.

Uma alternativa para realizar o monitoramento é através da utilização de imagens e vídeos. Com o desenvolvimento tecnológico, surgiram câmeras digitais de diversos tamanhos e resoluções, que podem ser adaptadas para utilização em situações nas quais o ser humano tem dificuldades (ou até impossibilidade) de acesso. Exemplos disso consistem na análise microscópica de materiais compósitos ou na análise mecânica de peças submetidas a altas temperaturas. As imagens captadas, em conjunto com a aplicação de técnicas de computação gráfica, oferecem ao usuário um conjunto extenso de informações.

Em computação gráfica, de uma maneira mais geral, são definidos processos que relacionam imagens e dados, conforme apresentado na Figura 1. A área de processamento de imagens opera realizando alterações de alguma maneira específica, geralmente visando a extração de uma característica. Um exemplo consiste na aplicação de filtros para destacar as bordas do contorno de um objeto.

Figura 1 – Relação entre áreas da computação gráfica.



Fonte: Autor.

A síntese de imagens consiste em técnicas de visualização de dados. São métodos usados para pós-processamento de resultados a fim de facilitar a compreensão do comportamento de um conjunto de informações. O processamento de dados abrange técnicas para filtragem de dados e extração de conhecimento (*data-mining*) de um conjunto de informações.

A área de análise de imagens se refere a métodos que extraem informações específicas (como reconhecimento de padrões, por exemplo) de um conjunto de imagens. A Correlação de Imagens está inserida nesse contexto como parte de uma subárea denominada Registro de Imagens. A técnica visa localizar em uma segunda imagem os pontos previamente definidos na imagem inicial. Na Correlação de Imagens, a busca por pontos correspondentes em duas imagens é medida por um índice de correlação.

Assim como o acompanhamento de um objeto em imagem é possível, pode-se também realizar o rastreamento de partes de um mesmo objeto. A busca envolve cálculo de correlações entre certas partes de diferentes imagens. Quando a correlação é executada sistematicamente em partes de um objeto submetido a variações de forma, é possível descobrir os deslocamentos internos ocorridos em determinados pontos daquele objeto. Ao obter um conjunto suficiente de pontos e seus respectivos deslocamentos, pode-se calcular a deformação local com razoável precisão.

De maneira geral, a correlação de imagens também permite que sejam calculadas outras grandezas físicas associadas aos materiais dos objetos estudados. Alguns exemplos consistem em um objeto que esteja sofrendo variações de temperatura cujo coeficiente de dilatação térmica deseja-se calcular; ou um objeto formado por material viscoelástico cujas propriedades se procura descobrir.

Assim, percebe-se o potencial da aplicação de técnicas de análise e processamento de imagens como uma forma alternativa de avaliação de grandezas físicas em ensaios. Do ponto de vista acadêmico, as pesquisas em técnicas de correlação de imagens digitais são cada vez mais variadas e apresentam-se nos mais diversos contextos.

1.1 Revisão de Literatura

O processo de correlação de imagens digitais (CID ou DIC do inglês *Digital Image Correlation*) faz parte de um contexto maior, onde se avalia o fluxo óptico de um objeto através da comparação entre duas imagens do objeto estudado em instantes de tempo diferentes.

Este trabalho trata do estudo de técnicas de correlação de pontos que se deslocaram dentro do plano da imagem (2D-DIC). No entanto, há outras vertentes dentro desta linha de pesquisa: tridimensionais (cujas direções dos deslocamentos não se resumem ao plano da imagem); utilização de duas câmeras (Stereo-DIC), conforme apresentado por Harvent et al. (2014); e o cálculo de correlação volumétrica digital (Digital Volume Correlation - DVC), como em Smith, Bay e Rashid (2002).

Dentro do contexto abordado neste trabalho, uma técnica de rastreamento bastante conhecida é apresentada por Lucas e Kanade (1981), na qual se calcula a correlação (C_{SSD}) através do somatório do quadrado da diferença entre as intensidades de cinza nos *pixels* das imagens avaliadas. Adotando o sistema de coordenadas da imagem original como $F(x; y)$ e o sistema de coordenadas da imagem deformada como $G(\tilde{x}; \tilde{y})$ e supondo uma imagem com dimensões $M \times N$ *pixels*, o coeficiente é calculado por

$$C_{SSD} = \sum_{i=1}^M \sum_{j=1}^N (F(x_i; y_j) - G(\tilde{x}_i; \tilde{y}_j))^2. \quad (1.1)$$

Por esse critério, quanto menor o valor de C_{SSD} , melhor é a correlação e mais parecidas são as imagens. As coordenadas da imagem deformada se relacionam com as coordenadas da imagem original por $\tilde{x} = x + u(x; y)$ e $\tilde{y} = y + v(x; y)$, onde as funções u e v definem o campo de deslocamentos nas direções X e Y, respectivamente.

A utilização do somatório do quadrado da diferença consiste em um dos critérios básicos para definição do coeficiente de correlação. Outro critério derivado de técnicas de processamento de sinais e utilizado nos primeiros artigos que desenvolveram a técnica é o de correlação cruzada. Os artigos de Peters et al. (1983), Chu, Ranson e Sutton (1985) e Sutton et al. (1986) são exemplos de aplicação desse tipo de coeficiente. O cálculo do coeficiente de correlação cruzada (C_{CC}) é realizado por

$$C_{CC} = \sum_{i=0}^M \sum_{j=0}^N (F(x_i; y_j)G(\tilde{x}_i; \tilde{y}_j)). \quad (1.2)$$

Nesse caso, e ao contrário de C_{SSD} , a correlação é melhor quando o valor de C_{CC} é máximo. A correlação de imagens foi inicialmente desenvolvida na Universidade da Carolina do Sul, nos Estados Unidos. Dentre alguns artigos dessa época, citam-se Peters et al. (1983), Sutton et al. (1983) e Chu, Ranson e Sutton (1985).

As primeiras versões dessa estratégia utilizavam variações das Equações 1.1 e 1.2 e as funções de mapeamento definiam transformações de primeiro grau (ou transformações afins). As funções de mapeamento usavam distâncias relativas Δx e Δy da coordenada em questão até o centro da imagem e parâmetros p_i de ponderação da seguinte forma:

$$u(x; y) = p_0 + p_2\Delta x + p_4\Delta y + p_6\Delta x\Delta y \quad (1.3)$$

$$v(x; y) = p_1 + p_3\Delta x + p_5\Delta y + p_7\Delta x\Delta y. \quad (1.4)$$

Os parâmetros p_0 e p_1 da função de mapeamento estão relacionados com translação de corpo rígido do objeto de estudo. Os outros parâmetros servem para definir deslocamentos de primeiro e segundo graus nas direções X e Y que ocorram no elemento estrutural.

As primeiras formas de cálculo desses parâmetros do campo de deslocamentos eram realizadas através de tentativa e erro. Era definido um possível intervalo de atuação do deslocamento ocorrido, testavam-se valores para o deslocamento de corpo rígido e registravam-se os coeficientes de correlação encontrados. O menor dos coeficientes definiria os valores dos dois primeiros parâmetros. Em seguida, eram definidos valores para os parâmetros p_2 e p_3 dentro do mesmo intervalo e encontravam-se novos valores dos coeficientes de correlação. Novamente, o menor coeficiente encontrado determinava os valores dos parâmetros. Esse processo continuava até que todos os parâmetros fossem calculados.

O método em questão possuía algumas desvantagens: caso o deslocamento fosse maior que o intervalo definido, o método não conseguiria encontrar uma resposta precisa. Além disso, a execução do método repetidas vezes era computacionalmente custosa.

De forma a atuar diretamente na automatização do processo, otimizando precisão e custo computacional, Bruck et al. (1989) apresentaram um procedimento de otimização não-linear baseado em Newton-Raphson para refino da solução. Em outro aspecto, Vendroux e Knauss (1998) desenvolveram uma simplificação a ser aplicada em Newton-Raphson para acelerar o desempenho da otimização.

A aplicação de um processo de otimização trouxe novas perspectivas para aplicação de outros métodos. Cheng et al. (2002), Shen e Paulino (2011) e Chen et al. (2013) usam como resolução o processo de Levenberg-Marquardt, que utiliza uma aproximação da matriz Hessiana construída a partir da matriz Jacobiana.

Os campos de deslocamentos são responsáveis pela correspondência entre as coordena-

das da imagem original e as coordenadas da imagem deformada. Geralmente, as coordenadas indicadas pelo campo de deslocamento apontam posições de ordem inferior a um *pixels*, que será denominada "subpixel". Os artigos clássicos de Sutton et al. (1983) e Chu, Ranson e Sutton (1985) utilizam interpolação bilinear para calcular a intensidade entre *pixels*.

Um avanço importante foi apresentado por Cheng et al. (2002), que utilizaram splines para fazer a interpolação em pontos *subpixélicos*. A utilização de splines diminui a ocorrência de inclinações divergentes, suavizando os valores das derivadas. Essa suavização permite que o método de análise usado encontre valores de deslocamentos mais consistentes em relação aos modelos reais. Schreier, Braasch e Sutton (2000) mostraram que a utilização de funções de maior grau, bem como aplicação de splines melhoram a precisão da posição do campo e da intensidade calculada.

Visando ao aumento da precisão dos campos, bem como a uma representação mais acurada de campos de deslocamentos heterogêneos, Lu e Cary (2000) propuseram a utilização de uma função mais elaborada, que incluía termos do segundo grau, para os campos de deslocamentos.

O processo de CID pode ser categorizado pelo tamanho da região de análise delimitada e pelas coordenadas de referência. Quando a região de análise abrange a maior parte da imagem, a abordagem de análise é global (ou *Full-Field*). Nesse caso, é definido um conjunto de pontos internos da região para o cálculo do coeficiente de correlação.

Por outro lado, a abordagem local (ou *Subset*) é usada quando se opta por um conjunto de regiões de análise significativamente menores do que a imagem. Nesse caso, um ponto central em cada região é usado como referência e todos os pontos internos da região de análise são usados no cálculo da correlação. De qualquer forma, a aplicação da estratégia de CID sempre está associada à minimização da correlação para pontos dentro de uma vizinhança.

O método *Full-Field* possui uma abordagem global de análise do campo de deslocamentos. Sua principal vantagem é manter a continuidade global do comportamento do objeto fazendo análises que mapeiam o campo de deslocamentos globalmente.

A desvantagem da estratégia *Subset* é a falta de informações sobre a continuidade entre campos de deslocamentos de diferentes células, o que pode ser uma fonte de erros na avaliação do campo de deformações. Por outro lado, sua principal vantagem consiste em uma maior velocidade de execução, já que as células utilizadas nas avaliações são menores.

Um dos primeiros artigos a citar a correlação por *Subset* foi apresentado por Peters et al. (1983), que delimitava células para execução da técnica. O artigo também apresentou dados sobre a influência da distância ao ponto central no coeficiente de correlação. Chu, Ranson e Sutton (1985) apresentaram a técnica de força-bruta, conforme descrita anteriormente.

Uma análise de erros sistemáticos do processo de CID foi realizada em Schreier, Braasch e Sutton (2000) e em Schreier e Sutton (2002). O trabalho de Schreier, Braasch e Sutton

(2000) analisou erros causados por funções de mapeamento que não conseguem definir o campo de deslocamentos real aplicado ao objeto de estudo. Em Schreier e Sutton (2002), foram analisados erros cometidos na interpolação das intensidades da região de análise. Ao longo dos anos, novas técnicas para correlação de imagens no modelo *Subset* foram propostas, resolvendo ou contornando as limitações apresentadas pela técnica.

Sun et al. (2005) e Besnard, Hild e Roux (2006), por exemplo, apresentaram associações entre o método *Subset* de correlação de imagens e o Método dos Elementos Finitos (MEF), de maneira a tornar a análise global. A associação é possível, pois a grade que divide a região de interesse pode ser interpretada como uma malha regular e a função de deslocamentos, como as funções de interpolações do elemento finito. Dessa maneira, pode-se utilizar essas funções de interpolação do MEF para manter a continuidade do campo de deslocamentos. Pan et al. (2015) compararam o desempenho entre o método *Subset* de abordagem local de análise com o método de CID com abordagem global usando funções de forma do MEF.

Outra abordagem com enfoque local semelhante ao *Subset* é apresentada nos artigos de Pan et al. (2009), Pan (2011) e Blaber, Adair e Antoniou (2015), que desenvolveram uma técnica de correlação de imagens baseada na confiabilidade dos elementos (*Reliability-Guided Digital Image Correlation* – RG-DIC). Essa técnica organiza os elementos de melhor correlação preliminar dentro de uma região de interesse e resolve-os seguindo essa ordem, de maneira que os elementos com melhor correlação sejam calculados inicialmente.

Ainda nessa direção, Jiang et al. (2014) e Blaber, Adair e Antoniou (2015) utilizaram um processo de otimização que realiza o cálculo da inversa da Hessiana através de uma composição baseada em Gauss-Newton (*Inverse Compositional Gauss-Newton* – IC-GN). Esse processo é aplicado à técnica RG-DIC. A otimização por IC-GN estima a inversa da Hessiana da imagem uma única vez e a utiliza durante todo o refino da solução, o que aumenta o desempenho computacional do método.

Em outro aspecto, é interessante ressaltar que o crescente desenvolvimento tecnológico tem permitido a fabricação de equipamentos que captam imagens com maior precisão, utilizando resoluções cada vez maiores. O aumento da quantidade de *pixels* por imagem, apesar de permitir melhores condições de avaliação a qualquer estudo realizado, compromete o desempenho computacional da análise. As funções de intensidade da imagem ficam maiores e, com isso, os métodos precisam analisar uma quantidade maior de informações.

Para contornar o problema do custo computacional causado pelo aumento do tamanho da imagem, uma solução pode ser a aplicação de técnicas computacionais de alto desempenho que acelerem o processo de refino da análise sem alterar a precisão do método. Atualmente, há trabalhos que realizam procedimentos de CID utilizando unidades de processamento gráfico (*Graphic Processing Unit* - GPU), como o trabalho de Leclerc et al. (2009), que usa o processamento em GPU para calcular propriedades mecânicas de materiais. No entanto, a aplicação de técnicas de computação de alto desempenho em soluções de CID é relativamente rara. Ainda

assim, várias soluções de análise por CID são encontradas na literatura.

Atualmente, as aplicações para CID são variadas e em crescente número. Dentre as aplicações, cita-se a identificação de parâmetros constitutivos para materiais. Grédiac (2004) apresentou aplicações para identificação de materiais compósitos, além de fazer uma revisão na área, e Avril et al. (2008) compararam cinco métodos recentemente desenvolvidos para identificação de propriedades anisotrópicas em materiais.

Os valores dos erros residuais sistemáticos são avaliados e há uma tendência para a padronização de indicadores e técnicas em metrologia óptica. Exemplos disso consistem na norma americana ASTM-E2208 (American Society for Testing and Materials, 2013) e no Glossário Internacional de Metrologia desenvolvido pelo Comitê Adjunto para Guias em Metrologia (Joint Committee For Guides In Metrology (JCGM), 2008).

Entre alguns artigos sobre quantificação de erros, citam-se Sutton et al. (2008), que demonstraram os efeitos das componentes dos deslocamentos que ocorrem fora do plano da imagem; e Wang e Pan (2015), que investigaram erros randômicos associados ao uso de funções de mapeamento coerentes com o campo de deslocamento aplicado.

São encontradas diversas soluções científicas e comerciais para o uso de CID. Como solução comercial, cita-se o VIC-2D (Correlated Solutions Inc., 2008). Dentre as soluções científicas, podem ser citadas o CorreliQ4, do Hild e Roux (2008) e o Ncorr, do Blaber, Adair e Antoniou (2015). A solução Ncorr é *open-source* e está disponibilizada em *site*.

A área de Registro de Imagens apresenta variados métodos para análise de imagens e extração de dados de interesse. Dentre esses métodos, um que se destaca é o *Scale Invariant Feature Transform* - SIFT, conforme descrito em Lowe (1999) e Lowe (2004).

O método busca na imagem características que não variam mesmo quando a imagem sofre translações, rotações, escala ou transformações afins. Assim, a busca por pontos correspondentes em diferentes imagens não sofre interferências de operações realizadas em imagens.

Dentro do Programa de Pós-Graduação em Engenharia Civil da Universidade Federal de Alagoas (PPGEC), a linha de pesquisa foi iniciada com Ferraz (2013). Ferraz (2013) calculou campos de deslocamentos e deformações para 5 (cinco) exemplos conhecidos na literatura utilizando um variado conjunto de células. Ferraz (2013) examinou, para cada um dos exemplos, os erros absolutos e relativos entre os campos de deslocamentos e deformações calculados pelo algoritmo e comparados com a literatura. Em seu algoritmo, foi usada a técnica de Newton-Raphson para otimização não-linear, além de interpolações bilineares e bicúbicas dentro do método *Subset*. Ferraz (2013) mostrou que o método apresentava maiores dificuldades no refinamento de campo de deslocamentos para estudos de casos mais complexos, como os ensaios de compressão triaxial e compressão diametral.

Tendências de aplicações para CID foram descritas em Sutton e Hild (2015). Uma área de interesse é o desenvolvimento de ferramentas que comparem simulações numéricas com

técnicas de CID para validação dos resultados. Um exemplo disso foi mostrado em Dufour et al. (2015). Outras tendências são de ligações entre tecnologias de aquisição de dados com técnicas de CID visando complementar as medições calculadas.

1.2 **Justificativa**

Conforme apresentado na revisão de literatura acima, diversas tecnologias têm sido incorporadas aos trabalhos de CID como alternativas viáveis para a melhoria da precisão e velocidade do cálculo dos campos de deslocamentos. Além disso, os métodos que suportam os processos de CID (interpolação, otimização não-linear, processamento de imagens e computação de alto desempenho) são cada vez mais variados, servindo para aplicações cada vez mais específicas.

No entanto, as soluções disponíveis na literatura apresentam poucas variações dos métodos de análise por CID e das técnicas que dão suporte a esse tipo de análise. Geralmente, as soluções apresentam um conjunto limitado de técnicas, que são utilizadas para um mesmo tipo de aplicação. Além disso, não se encontrou registro de uma arquitetura formalizada de um ambiente computacional que permita a análise de campos de deslocamentos através da CID.

Neste sentido, identifica-se a necessidade de apresentação de uma arquitetura conceitual que integre diversas tecnologias usadas para o cálculo de grandezas de engenharia através da aplicação de técnicas de correlação de imagens digitais. O desenvolvimento de uma arquitetura conceitual integra um conjunto abrangente de estratégias e métodos usados em CID em um ambiente computacional único e de alto desempenho. Assim, a arquitetura consiste em um primeiro passo para o desenvolvimento de um ambiente computacional capaz de ser utilizado em várias aplicações.

1.3 **Objetivos e Metas**

O objetivo principal do trabalho consiste na apresentação de um ambiente computacional de alto desempenho que utiliza técnicas de correlação de imagens digitais para determinação de grandezas físicas associadas a problemas de engenharia. O ambiente computacional é produto de uma arquitetura que integra diversas tecnologias avançadas e usadas no cálculo dos campos de deslocamentos de elementos estruturais usando dados extraídos de ensaios não-destrutivos.

O objetivo geral é alcançado com o atendimento das seguintes metas:

1. Estudo e análise de técnicas de correlação de imagens digitais, interpolação bidimensional, estratégias de otimização não-linear e métodos de computação de alto desempenho;
2. Planejamento do modelo conceitual, com análise de técnicas de Arquitetura de *Software*

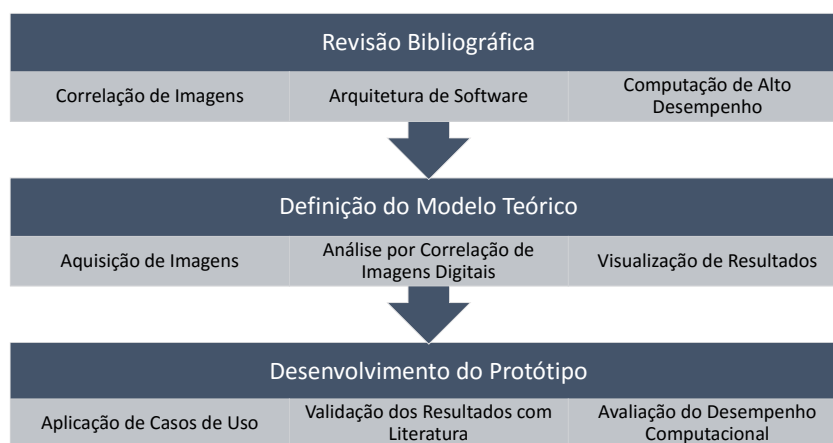
e padrões de projeto (*Design Pattern*), juntamente com o uso de ferramentas para o desenvolvimento inicial da arquitetura;

3. Integração das tecnologias e estratégias utilizadas em correlação de imagens digitais ao modelo teórico desenvolvido e adequação do ambiente conceitual aos métodos de análise;
4. Desenvolvimento de um protótipo do ambiente computacional de alto desempenho apresentado, permitindo a execução de cálculos de deslocamentos e deformações a partir da correlação de imagens digitais;
5. Testes de estudos de caso no protótipo do ambiente de alto desempenho para:
 - a) Validação dos resultados calculados pelo protótipo, comparados com valores analíticos obtidos na literatura;
 - b) Validação do aumento da performance computacional do protótipo ao incorporar técnicas de computação de alto desempenho.
6. Uso do método SIFT para comparação com o DIC como um cálculo alternativo dos parâmetros dos campos de deslocamentos.

1.4 Metodologia

Para alcançar os objetivos propostos neste trabalho, a metodologia de desenvolvimento é dividida em três (3) grandes etapas, conforme ilustrado no fluxograma (Figura 2).

Figura 2 – Fluxograma da Pesquisa.



Fonte: Autor.

A atividade de revisão bibliográfica envolve o estudo de formulações e métodos de correlação de imagens digitais e computação de alto desempenho. O estudo de estratégias de cálculo do campo de deslocamentos e deformações por DIC exige conhecimento de técnicas de processamento de imagens, otimização não-linear e interpolações bidimensionais. Nesta etapa

também é realizado o estudo de Arquitetura de Software e modelos de padrão de projetos para posterior desenvolvimento da arquitetura do ambiente computacional de alto desempenho.

A segunda parte deste trabalho corresponde ao desenvolvimento da arquitetura do ambiente computacional de alto desempenho. A arquitetura representa um modelo conceitual que integra diversas tecnologias comuns aos procedimentos de análise por CID. Uma das tarefas iniciais corresponde ao planejamento da arquitetura do ambiente computacional apresentado. Nesta etapa, é realizada uma identificação dos procedimentos comuns aos métodos de correlação, separando esses processos em módulos que caracterizam cada fase importante da análise computacional. A arquitetura se divide em três módulos: aquisição das imagens, análise usando correlação de imagens digitais e visualização dos resultados. A incorporação de funções de mapeamento para os campos de deslocamentos é realizada no módulo de análise.

A próxima etapa da metodologia consiste no desenvolvimento de um protótipo do ambiente computacional de alto desempenho. Esse protótipo, desenvolvido para fins de validação da arquitetura, permite a execução do procedimento de cálculo de campos de deslocamentos em imagens através da CID.

A última etapa da metodologia valida ainda o protótipo desenvolvido através da aplicação de estudos de casos analíticos encontrados na literatura. São realizadas análises comparativas entre os resultados obtidos pelo protótipo e os valores analíticos dos modelos obtidos na literatura. Também é avaliado o ganho em performance computacional obtido com a incorporação de técnicas de computação de alto desempenho.

A inclusão do método SIFT no presente trabalho objetiva a comparação entre métodos de registro de imagens, bem como validação dos resultados obtidos pelas duas técnicas. Os resultados obtidos usando SIFT serão comparados com os resultados obtidos usando DIC. Ambos serão comparados com os valores de referência dos modelos da transformação, desenvolvido a partir de exemplos da literatura.

1.5 **Delimitação do Trabalho**

O foco do presente trabalho consiste na definição da arquitetura computacional do módulo de análise. Também objetiva desenvolver métodos do módulo de análise para o protótipo do ambiente computacional de alto desempenho. Dos métodos de análise, será desenvolvida a técnica *Subset*, de abordagem local.

Assim, não será definida a arquitetura para os módulos de aquisição de imagens digitais e visualização de resultados na arquitetura conceitual. Também não será realizado o desenvolvimento desses módulos no protótipo do ambiente computacional de alto desempenho. Dentro do módulo de análise, não serão estudados métodos de abordagem global (*Full-Field*).

1.6 **Escopo do Trabalho**

O Capítulo 1 apresenta a introdução ao trabalho, com apresentação da revisão bibliográfica do assunto, apresentação da justificativa para execução do projeto, objetivos e metas a serem atendidos pelo projeto, metodologia de execução e delimitação do projeto. O Capítulo 2 define alguns conceitos básicos necessários ao entendimento da pesquisa, bem como uma explicação das técnicas usadas para execução da correlação entre imagens digitais. O Capítulo 3 apresenta o modelo conceitual do ambiente computacional a ser desenvolvido, com definição dos módulos de implementação planejados e do fluxograma do processo de análise. O Capítulo 4 descreve a implementação do protótipo de análise desenvolvido, com informações sobre os métodos usados e as técnicas implementadas. O Capítulo 5 contém os tipos de análises realizadas, juntamente com as imagens geradas para cálculo dos deslocamentos. Os resultados encontrados são comparados com modelos de transformação usados na geração de imagens. Também são apresentados resultados provenientes de outras técnicas de análise de imagens, que são comparadas com os resultados calculados pelo protótipo. O Capítulo 6 apresenta as conclusões encontradas no desenvolvimento do presente trabalho, além de um conjunto de sugestões de trabalhos futuros.

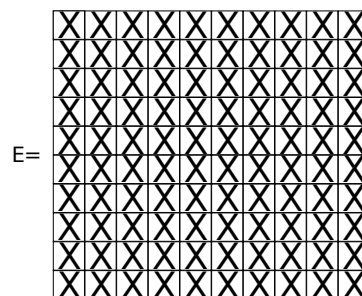
2 CONCEITOS BÁSICOS

A seguir, serão apresentados conceitos relativos ao processo de correlação de imagens digitais e computação de alto desempenho. Para maiores informações sobre o assunto, recomendam-se Schreier, Orteu e Sutton (2009), Grédiac, Hild e Pineau (2012) e Ferraz (2013)

2.1 Análise de Imagens Digitais

O conceito de imagem digital, segundo Banon (2000), associa um domínio espacialmente dividido a um domínio de cores. Sejam dois conjuntos distintos E e K , onde E é um conjunto de quadrados adjacentes arranjados ao longo de m linhas e n colunas e o domínio K , um domínio de cores. Especificamente para a correlação, opta-se por um conjunto de intensidades de cinza. A Figura 3 apresenta um exemplo de domínio com 10 linhas e 10 colunas vazias e a Figura 4 mostra um exemplo de um conjunto representando um domínio com quatro intensidades de cinza.

Figura 3 – Domínio espacial.



Fonte: (BANON, 2000).

Figura 4 – Domínio de intensidades.



Fonte: (BANON, 2000).

Banon (2000) diz que uma imagem consiste em uma relação cartesiana entre o domínio espacial E e o domínio de intensidades K de forma que, para cada quadrado pertencente ao domínio E , exista apenas uma intensidade de cinza pertencente ao domínio K . A Análise de Imagens é realizada em imagens assim definidas.

Segundo Brown (1992), Registro de Imagens é o processo de análise de duas ou mais imagens de uma mesma cena tirada em diferentes momentos, por diferentes pontos de vista e/ou por diferentes tipos de sensores para superposição das imagens.

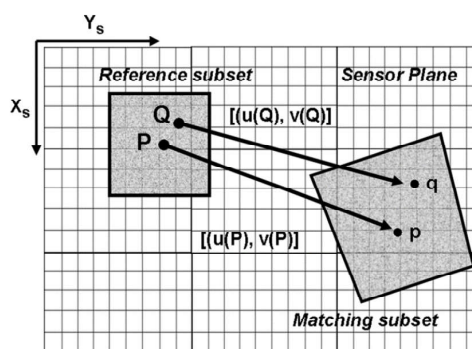
Complementando Brown (1992), Zitová e Flusser (2003) apresentam o processo de análise de imagens em quatro etapas: a detecção da característica, a correspondência da característica em questão, o cálculo do modelo de transformação (ou função de mapeamento) e a reconstrução da imagem usando a função de mapeamento calculada.

Nessa divisão, os métodos de correlação executam a correspondência entre características específicas da imagem. A característica é um conjunto de *pixels* da imagem no instante original a ser procurado na imagem em um outro momento. A função de mapeamento caracteriza a transformação que ocorre na imagem e os parâmetros do modelo são calculados a partir de métodos de otimização não-linear. A otimização visa encontrar um máximo/mínimo para o coeficiente de correlação adotado.

2.2 Correlação de Imagens Digitais

Zitová e Flusser (2003) enquadram a correlação como um método baseado em área da imagem e usa uma métrica para a semelhança entre regiões de interesse (ROI) em imagens diferentes, como ilustrado na Figura <>. Atualmente, duas expressões (e suas variações) se destacam: a correlação cruzada (CC) e o somatório do quadrado da diferença (SSD) entre imagens. Para a correlação cruzada, quanto maior o valor, maior a semelhança entre duas ROI. No caso do somatório do quadrado da diferença, quanto mais próximo do valor nulo, melhor é a correlação entre duas imagens.

Figura 5 – Representação da correlação de imagens digitais.



Fonte: Schreier, Orteu e Sutton (2009).

A correlação cruzada é um método baseado em técnicas de processamento de sinais, onde há medição da semelhança entre dois sinais que estejam defasados por um tempo t através da convolução entre os sinais estudados. O operador convolução entre dois sinais f e g é

definido

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau. \quad (2.1)$$

As fórmulas para cálculo dos coeficientes de correlação apresentam variações mais elaboradas, desenvolvidas para diminuir ruídos comumente encontrados durante a aquisição de imagens digitais de objetos reais. Entre esses ruídos, pode-se destacar a distorção da lente da câmera e mudanças na iluminação durante o processo de captação.

Variações da fórmula de correlação cruzada são apresentadas a seguir. A correlação cruzada normalizada (*Normalized Cross-Correlation*) visa diminuir problemas de mudança de escala na imagem (PAN, 2011), e é dada por

$$C_{NCC} = \sum_{i=1}^M \sum_{j=1}^M \frac{F(x_i; y_j)G(\tilde{x}_i; \tilde{y}_j)}{\bar{F} \bar{G}} \quad (2.2)$$

onde

$$\bar{F} = \sqrt{\sum_{i=1}^M \sum_{j=1}^M [F(x_i; y_j)]^2} \quad (2.3a)$$

$$\bar{G} = \sqrt{\sum_{i=1}^M \sum_{j=1}^M [G(\tilde{x}_i; \tilde{y}_j)]^2}. \quad (2.3b)$$

Outra variação da correlação cruzada consiste na fórmula com normalização em zero (*Zero Normalized Cross-Correlation*). Segundo Pan (2011), essa normalização, além de diminuir os efeitos de escala da imagem, também diminui os ruídos provenientes da iluminação. Essa correlação é dada por

$$C_{ZNCC} = \sum_{i=1}^M \sum_{j=1}^M \frac{[F(x_i; y_j) - f_m][G(\tilde{x}_i; \tilde{y}_j) - g_m]}{\Delta F \Delta G} \quad (2.4)$$

onde

$$f_m = \frac{1}{(2M+1)^2} \sum_{i=-M}^M \sum_{j=-M}^M F(x_i; y_j) \quad (2.5a)$$

$$g_m = \frac{1}{(2M+1)^2} \sum_{i=-M}^M \sum_{j=-M}^M G(\tilde{x}_i; \tilde{y}_j) \quad (2.5b)$$

$$\Delta F = \sqrt{\sum_{i=-M}^M \sum_{j=-M}^M [F(x_i; y_j)]^2} \quad (2.5c)$$

$$\Delta G = \sqrt{\sum_{i=-M}^M \sum_{j=-M}^M [G(\tilde{x}_i; \tilde{y}_j)]^2}. \quad (2.5d)$$

Conforme explicado no início da seção, a outra expressão utilizada para o cálculo do coeficiente de correlação é o somatório do quadrado da diferença das intensidades (*Sum of Squared Differences - SSD*), definida por

$$C_{SSD} = \sum_{i=-M}^M \sum_{j=-M}^M [F(x_i; y_j) - G(\tilde{x}_i; \tilde{y}_j)]^2 \quad (2.6)$$

Essa expressão apresenta variações semelhantes àquelas apresentadas para a fórmula de correlação cruzada, dadas por

$$C_{NSSD} = \sum_{i=-M}^M \sum_{j=-M}^M \left[\frac{F(x_i; y_j)}{\bar{F}} - \frac{G(\tilde{x}_i; \tilde{y}_j)}{\bar{G}} \right]^2 \quad (2.7a)$$

$$C_{ZNSSD} = \sum_{i=-M}^M \sum_{j=-M}^M \left[\frac{F(x_i; y_j) - f_m}{\Delta F} - \frac{G(\tilde{x}_i; \tilde{y}_j) - g_m}{\Delta G} \right]^2 \quad (2.7b)$$

onde os valores de $\bar{F}, \bar{G}, f_m, g_m, \Delta F$ e ΔG são os mesmos apresentados anteriormente nas Equações 2.3 e 2.5.

O processo de análise usando CID pode ser realizado a partir de duas abordagens diferentes. Cada abordagem adotada para o cálculo da correlação, associada à função campo de deslocamentos, apresenta um conjunto de vantagens e desvantagens, conforme descrito na Seção seguinte. As abordagens podem ser locais ou globais e estão definidas na seção seguinte.

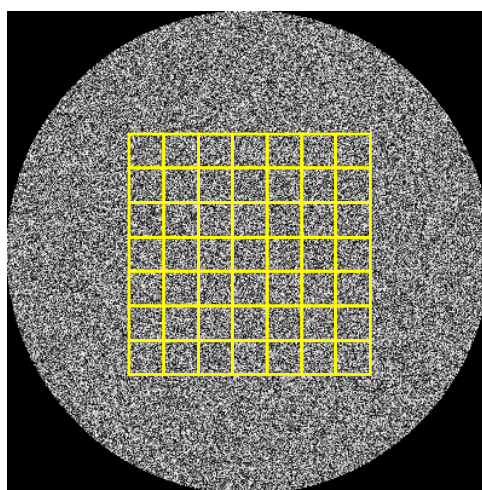
2.3 Abordagem Global/Local de Análise

Em termos gerais, a estratégia de análise define o tipo de abordagem adotada para execução do processo de cálculo dos campos de deslocamentos por CID. As estratégias mais comuns são denominadas *Full-Field*, de abordagem global, e *Subset*, de abordagem local.

A estratégia *Subset* consiste na divisão da região de interesse da imagem em células menores, onde cada uma possui um ponto central que serve de referência para a análise por CID. Nesse caso, os campos de deslocamentos são calculados para cada uma das células. As funções dos campos de deslocamentos apresentam parâmetros relacionados com a distância ao ponto central da célula, dentro do sistema de coordenadas local. Como o tamanho das células é reduzido, a velocidade de execução do processo por célula é alta.

O processo de correlação de cada célula é realizado separadamente, o que permite a aplicação direta de técnicas de computação em alto desempenho. Nesse sentido, a aplicação de técnicas de computação de alto desempenho é realizada de maneira simplificada. Em casos onde a imagem apresenta resolução alta e uma grande quantidade de células, o uso de técnicas para computação de alto desempenho é imprescindível. A Figura 6 ilustra uma grade usada seguindo a estratégia *Subset*.

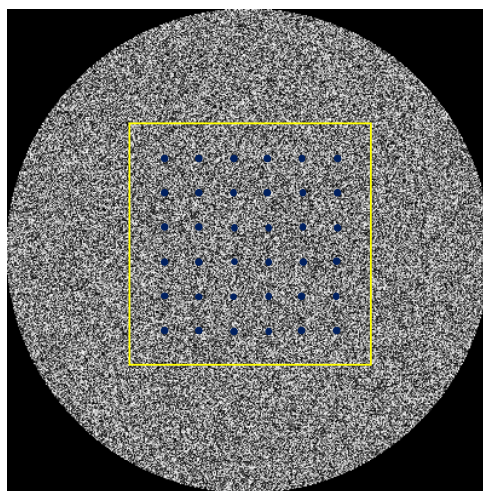
Figura 6 – Grade de células para análise seguindo o método *Subset*.



Fonte: Autor.

A principal desvantagem da estratégia *Subset* decorre da divisão da região de interesse em janelas independentes. Com isso, perde-se a informação de continuidade entre campos de deslocamentos pertencentes a células vizinhas. Assim, visando resolver o problema da falta de continuidade entre os campos, foi desenvolvida a estratégia *Full-Field*.

O método *Full-Field* se resume a uma abordagem global, onde a região de interesse delimitada abrange grande parte da imagem. Dentro dessa região, define-se uma malha de pontos que são usados no cálculo do coeficiente de correlação. A Figura 7 exemplifica uma região de interesse e a malha de pontos usada para o cálculo do coeficiente *Full-Field*.

Figura 7 – Grade de pontos para análise seguindo o método *Full-Field*.

Fonte: Autor.

O método *Full-Field* apresenta também como diferença a função campo de deslocamentos, que é definida para toda região, ao contrário do campo usado na abordagem local, que refere-se apenas à célula estudada. Assim como no método *Subset*, a correção dos valores dos parâmetros é feita usando um método de otimização não-linear.

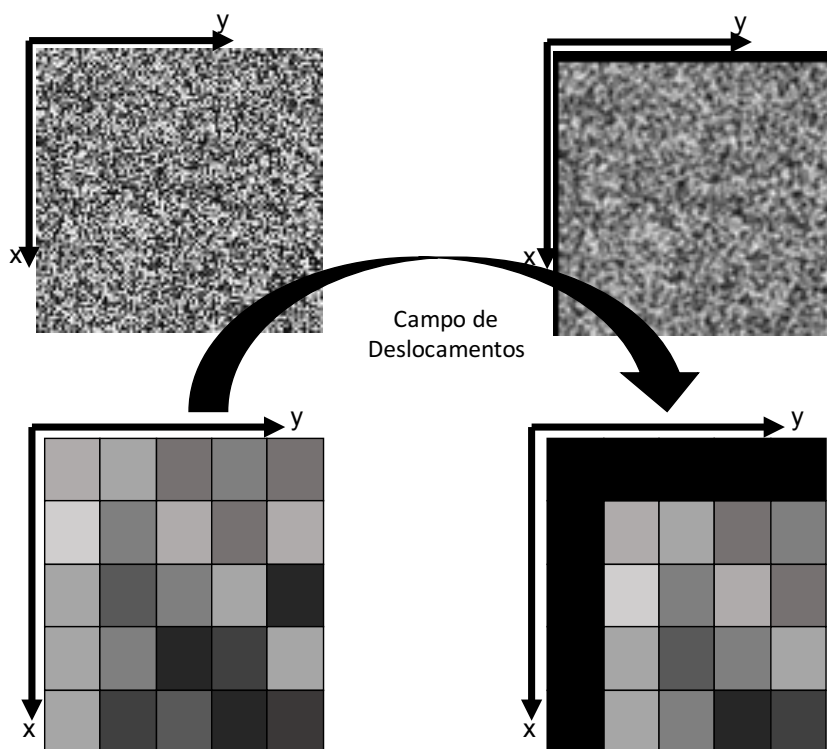
Para que a correlação entre duas imagens possa ser realizada, é interessante que se use uma função que relacione o sistema de coordenadas da imagem original, com o sistema de coordenadas da imagem deformada. Essa correspondência de coordenadas é feita através da função campo de deslocamentos, cujos parâmetros constituem as principais incógnitas do processo de correlação.

2.4 Funções de Mapeamento/Modelo de Transformação

O coeficiente de correlação é calculado usando um somatório que envolve as intensidades dos pontos da imagem original e seus respectivos pontos na imagem deformada. A obtenção das coordenadas do ponto na imagem deformada é realizada através da aplicação de funções de mapeamento do campo de deslocamentos nas coordenadas do ponto da imagem original.

O modelo de transformação da imagem é sensível ao tipo de abordagem adotada para análise. O modelo global usa pontos característicos previamente definidos para encontrar um modelo que se adeque a toda imagem, enquanto o modelo local trata a imagem como uma composição de partes (tesselação). A Figura 8 ilustra uma representação da função campo de deslocamentos.

Figura 8 – Função do campo de deslocamentos.



Fonte: Autor.

O campo de deslocamentos relaciona o sistema de coordenadas da imagem original com o sistema de coordenadas da imagem deformada. Para exemplificar, a Equação 2.8 apresenta uma função campo de deslocamentos polinomial:

$$\tilde{x} = x + \sum_{k=0}^N p_{2k} \psi_i \quad (2.8a)$$

$$\tilde{y} = y + \sum_{k=0}^N p_{2k+1} \psi_i \quad (2.8b)$$

onde \tilde{x}/\tilde{y} representa o sistema de coordenadas da imagem deformada, x/y indica o sistema de coordenadas da imagem original, p_i são os parâmetros a serem otimizados para encontrar a melhor correlação, ψ_i são as funções que compõem o campo de deslocamentos e N é o número de funções utilizadas para compor o campo.

Em geral, as funções ψ_i utilizadas no campo de deslocamentos dependem dos eixos globais do sistema de coordenadas da imagem ou estão relacionadas com a distância da origem ao sistema de coordenadas. A diferença na composição do campo de deslocamentos define a abordagem a ser usada para a análise por CID.

O caso mais simples de uma representação do campo corresponde ao deslocamento

de corpo rígido, onde todos os *pixels* são submetidos a uma translação linear horizontal e/ou vertical de determinada distância. O campo pode ser representado como a equação 2.9

$$x_d = x_0 + p_0 \quad (2.9a)$$

$$y_d = y_0 + p_0. \quad (2.9b)$$

onde x_d e y_d são as coordenadas dos *pixels* na imagem deformada e x_0 e y_0 são as coordenadas dos *pixels* na imagem original. Os parâmetros p_0 e p_1 são os coeficientes de translação aplicados na imagem original. Percebe-se que as funções ψ_i do campo de deslocamentos correspondem à unidade. O processo de cálculo dos parâmetros em situações onde se sabe que não houve nenhum tipo de deformação pode ser realizado usando o campo acima descrito.

Outro caso bastante utilizado consiste em transformações de primeiro grau. Para os casos mais simples, como tração uniaxial, a aplicação de parâmetros de primeiro grau são suficientes para caracterização do comportamento físico. Os campos, então, podem seguir a expressão

$$x_d = x_0 + p_0 + p_2\Delta x + p_4\Delta y \quad (2.10a)$$

$$y_d = y_0 + p_1 + p_3\Delta x + p_5\Delta y. \quad (2.10b)$$

A representação em funções de mapeamento do segundo grau permite uma combinação ainda maior de possíveis deslocamentos. A inclusão de parâmetros de segundo grau facilita o cálculo de campos de deslocamentos ainda mais elaborados, onde os campos de deformações atuantes na imagem podem ser lineares. Um campo de deslocamentos de segundo grau

$$x_d = x_0 + p_0 + p_2\Delta x + p_4\Delta y + \frac{1}{2}p_6\Delta x^2 + \frac{1}{2}p_8\Delta y^2 + p_{10}\Delta x\Delta y \quad (2.11a)$$

$$y_d = y_0 + p_1 + p_3\Delta x + p_5\Delta y + \frac{1}{2}p_7\Delta x^2 + \frac{1}{2}p_9\Delta y^2 + p_{11}\Delta x\Delta y \quad (2.11b)$$

pode representar um campo de flexão pura, por exemplo.

Outras representações de funções de mapeamento usam parâmetros associados ao modelo de câmera adotado, em caso de projeção ortográfica ou em perspectiva. As funções de mapeamento com parâmetros a_i , b_i e c_i e coordenadas x e y da imagem, segundo Zitová e Flusser (2003), consideram modelo de projeção em perspectiva

$$u = \frac{a_0 + a_1x + a_2y}{1 + c_1x + c_2y} \quad (2.12a)$$

$$v = \frac{b_0 + b_1x + b_2y}{1 + c_1x + c_2y}. \quad (2.12b)$$

Na literatura, encontram-se variados tipos de funções para compor o campo de deslocamentos. O processo de minimização do coeficiente de correlação usa técnicas de otimização não-linear, que calcula os valores dos parâmetros de tal forma que o coeficiente de correlação seja mínimo. O processo matemático para otimização não-linear por CID é discutido na seção seguinte.

2.5 Otimização Não-Linear

A determinação dos parâmetros das funções de mapeamento utilizando técnicas de correlação de imagens digitais envolve a aplicação de métodos de otimização não-linear pois a função objetivo não é linear. O campo de deslocamentos não-linear é ajustado iterativamente.

O coeficiente de correlação de imagens se resume a um indicador usado para medir a semelhança entre duas imagens. O cálculo é uma das ferramentas do processo de cálculo do campo de deslocamentos e deformações. Uma das partes mais importante é a aplicação de um conjunto de ferramentas matemáticas que permitam a utilização de valores que ponderem o campo de deslocamentos, minimizando o coeficiente de correlação.

O processo de otimização existe como uma técnica alternativa usada na obtenção dos campos de deslocamentos. Sua incorporação no processo garante que, para certas condições, encontre-se um conjunto de parâmetros que represente o comportamento físico do objeto de estudo.

Tomando como expressão para o coeficiente de correlação o somatório do quadrado da diferença (SSD), conforme apresentado na Equação 2.13, pode-se avaliar a fórmula por um enfoque diferenciado. A diferença entre a intensidade de um ponto na imagem original e seu provável correspondente, definido pelo campo de deslocamentos, pode ser visto como um resíduo: um valor que indica o erro existente pela não adequação do campo de deslocamentos.

$$C_{SSD} = \sum_{i=1}^M \sum_{j=1}^M [F(x_i; y_j) - G(\tilde{x}_i; \tilde{y}_j)]^2 \quad (2.13)$$

onde $F(x; y)$ representa a função da imagem original e $G(\tilde{x}_i; \tilde{y}_j)$ a função da imagem deformada, utilizando um sistema de coordenadas diferente do sistema de coordenadas da imagem original. A relação entre os dois sistemas é definida pela função campo de deslocamentos.

Com isso, o processo de otimização também pode ser entendido como uma minimização generalizada dos resíduos calculados. A escolha por minimizar o quadrado dos resíduos ao invés de utilizar a função módulo acontece pois a função módulo não é diferenciável na origem, o que impede a utilização de métodos que envolvam diferenciações, como o método de Newton-Raphson, por exemplo.

Matematicamente, o processo de minimização envolve encontrar o valor onde o gradiente apresenta valor nulo. O cálculo dos valores das primeiras derivadas (gradiente), é realizado

derivando-se a Equação 2.6 em relação aos parâmetros associados ao campo de deslocamentos. A derivada $\partial C(x; y; \mathbf{P})/\partial \mathbf{P}$ da Equação 2.13 em relação ao vetor \mathbf{P} de parâmetros da função de mapeamento é

$$\nabla \mathbf{C} = \frac{\partial C}{\partial \mathbf{P}} = -2 \left[\sum_{i=1}^M \sum_{j=1}^M (F(x_i; y_j) - G(\tilde{x}_i; \tilde{y}_j)) \frac{\partial G(\tilde{x}_i; \tilde{y}_j; \mathbf{P})}{\partial \mathbf{P}} \right] = \mathbf{0}. \quad (2.14)$$

Para encontrar os valores das raízes de $\nabla \mathbf{C} = \mathbf{0}$, resolve-se recursivamente a Equação 2.15 representada na sequência e obtida pela simplificação da expressão em termos da série de Taylor, e onde P_0 indica uma estimativa inicial para otimização.

$$\nabla \nabla C(\mathbf{P})(\mathbf{P} - \mathbf{P}_0) = -[\nabla C(\mathbf{P}_0)] \quad (2.15)$$

A segunda derivada $\nabla \nabla C$ da expressão do coeficiente de correlação, denominada Hessiana da expressão, é obtida com

$$\begin{aligned} \nabla \nabla C(\mathbf{P}) &= \left(\frac{\partial^2 C}{\partial \mathbf{P} \partial \mathbf{P}} \right) \\ &\approx \left[-2 \sum_{i=1}^M \sum_{j=1}^M (F(x_i; x_j) - G(\tilde{x}_i; \tilde{y}_j)) \frac{\partial^2 C}{\partial \mathbf{P} \partial \mathbf{P}} \right. \\ &\quad \left. + 2 \sum_{i=1}^M \sum_{j=1}^M \frac{\partial G(\tilde{x}_i; \tilde{y}_j; \mathbf{P})}{\partial \mathbf{P}} \frac{\partial G(\tilde{x}_i; \tilde{y}_j; \mathbf{P})}{\partial \mathbf{P}} \right]. \end{aligned} \quad (2.16)$$

Vendroux e Knauss (1998) apresentaram uma simplificação para o cálculo da matriz Hessiana descrita acima. Quando os valores dos parâmetros do campo de deslocamentos são muito próximos do valor correto, $F(x; x) \approx G(\tilde{x}; \tilde{y})$.

Vendroux e Knauss (1998) mostraram resultados onde, caso a solução real esteja dentro de um raio de sete (7) *pixels*, a suposição acima pode ser usada e a primeira parte do cálculo da Hessiana pode ser desprezada, ficando a expressão

$$\nabla \nabla C(\mathbf{P}) \approx 2 \sum_{i=-M}^M \sum_{j=-M}^M \frac{\partial G(\tilde{x}; \tilde{y}; \mathbf{P})}{\partial \mathbf{P}} \frac{\partial G(\tilde{x}; \tilde{y}; \mathbf{P})}{\partial \mathbf{P}}. \quad (2.17)$$

As derivadas da intensidade do cinza em relação aos parâmetros do mapeamento do campo de deslocamentos são funções do campo de deslocamentos (necessário para a integração entre imagens original e deformada) e da interpolação utilizada para o cálculo da intensidade de um ponto subpixélico. As derivadas parciais podem ser calculadas a partir da regra da cadeia. A Equação 2.18 apresenta a derivadas parciais calculadas.

$$\frac{\partial G(\tilde{x}; \tilde{y}; \mathbf{P})}{\partial \mathbf{P}} = \frac{\partial G(\tilde{x}; \tilde{y}; \mathbf{P})}{\partial \tilde{x}} \frac{\partial \tilde{x}(x; y)}{\partial \mathbf{P}} + \frac{\partial G(\tilde{x}; \tilde{y}; \mathbf{P})}{\partial \tilde{y}} \frac{\partial \tilde{y}(x; y)}{\partial \mathbf{P}} \quad (2.18)$$

Nessa equação, as derivadas $\partial G(\tilde{x}; \tilde{y})/\partial \tilde{x}$ e $\partial G(\tilde{x}; \tilde{y})/\partial \tilde{y}$ representam as derivadas parciais da função interpoladora pelas coordenadas \tilde{x} e \tilde{y} do sistema de coordenadas da imagem deformada. As derivadas $\partial \tilde{x}(x; y)/\partial \mathbf{P}$ e $\partial \tilde{y}(x; y)/\partial \mathbf{P}$ são as derivadas parciais da função campo de deslocamentos pelos seus parâmetros de ponderação.

O método de otimização não-linear procura, na imagem deformada, as coordenadas que minimizem o coeficiente de correlação para os pontos correspondentes na imagem original. Durante a procura, o método modifica os parâmetros dos campos de deslocamentos e esses campos, geralmente, apontam posições entre *pixels*. Como a imagem é um conjunto discreto de tons de cinza, surge a necessidade de interpolação para determinação de intensidades subpixélicas.

2.6 Interpolação

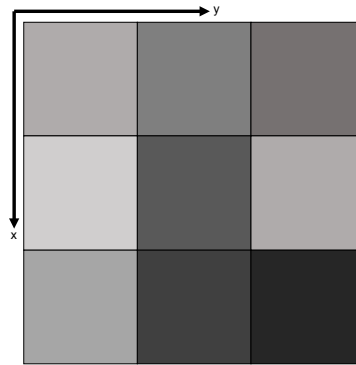
A utilização de técnicas de interpolação no processo de minimização do coeficiente de correlação usando CID é imprescindível pois, com a aplicação do valor da coordenada no campo de deslocamentos, o ponto correspondente, inserido na imagem deformada, não possui coordenadas inteiras na maioria dos casos. É importante lembrar que uma imagem digital é, basicamente, uma matriz com valores de intensidades da cor, onde cada posição é definida por um valor inteiro.

A correspondência entre as imagens original e deformada é feita através da função de mapeamento. Há casos onde a função de mapeamento retorna uma posição entre os *pixels*. Assim, para encontrar a intensidade da cor nessa posição, interpola-se usando as coordenadas e o valor das intensidades nos *pixels* vizinhos. Dependendo da interpolação adotada, calculam-se também os valores das derivadas das intensidades de *pixels* vizinhos.

Dentre as funções que podem ser usadas para interpolação bidimensional, ressalta-se o uso das *splines*. As funções dessa classe se adequam não apenas aos valores das intensidades nas imagens, mas apresentam em sua formulação tratamentos para adaptação aos valores das derivadas. Dessa forma, a função apresenta continuidade e é bem-comportada.

Toma-se como exemplo a formulação de uma *spline* bidimensional de grau 3 (*spline* bicúbica), conforme apresentada por Ferraz (2013). Utiliza-se uma célula 3x3 onde são obtidos os valores da intensidade nos pontos, bem como os valores das primeiras derivadas em relação aos dois eixos e sua derivada cruzada. A Figura 9 apresenta as células utilizadas na interpolação.

Figura 9 – Subcélula utilizada em interpolação 3x3.



Fonte: Autor.

O cálculo das derivadas em relação aos eixos e a derivada cruzada pode ser realizado utilizando o Método das Diferenças Finitas, que necessita das células vizinhas para execução dos cálculos. A interpolação por *splines* $g(h, t)$ utilizam os coeficientes de ponderação α_{mn} e as coordenadas relativas h e t para uma célula como

$$g(\tilde{x}; \tilde{y}) = \sum_{m=0}^3 \sum_{n=0}^3 \alpha_{mn} h^m t^n \quad (2.19a)$$

$$g_x = g_h(h, t) = \sum_{m=0}^3 \sum_{n=0}^3 m \alpha_{mn} h^{m-1} t^n \quad (2.19b)$$

$$g_y = g_t(h, t) = \sum_{m=0}^3 \sum_{n=0}^3 n \alpha_{mn} h^m t^{n-1} \quad (2.19c)$$

$$g_{xy} = g_{ht}(h, t) = \sum_{m=0}^3 \sum_{n=0}^3 mn \alpha_{mn} h^{m-1} t^{n-1}. \quad (2.19d)$$

As quatro expressões, para cada um dos pontos, servem para montar um sistema de equações lineares, cujo vetor solução são os valores das intensidades e das derivadas de cada ponto na Figura 9

$$[A] \{\alpha_{mn}\} = \left\{ \begin{array}{c} g^1 \\ \dots \\ g_x^1 \\ \dots \\ g_y^1 \\ \dots \\ g_{xy}^1 \end{array} \right\}. \quad (2.20)$$

O vetor das incógnitas contém os coeficientes α_{mn} das *splines* interpoladas que, para o caso da função cúbica, possui 16 coeficientes. Para fórmulas de maior grau, são necessários

mais coeficientes: *splines* de grau 5 precisam de 36 coeficientes e grau 7 utiliza 64 coeficientes, por exemplo.

2.7 Computação de Alto Desempenho

A aplicação de computação de alto desempenho (CAD) em técnicas de correlação de imagens digitais é uma atividade relativamente nova. O principal objetivo da aplicação de CAD consiste na distribuição de análises, de maneira que mais de um processador (ou computador) realizem os cálculos necessários à minimização do coeficiente de correlação.

Uma forma de se classificar as técnicas de CAD decorre da existência de compartilhamento de memória entre processadores. Técnicas de memória compartilhada são geralmente usadas quando um único computador possui vários processadores. Nesse caso, todos os processadores têm acesso ao mesmo barramento de memória.

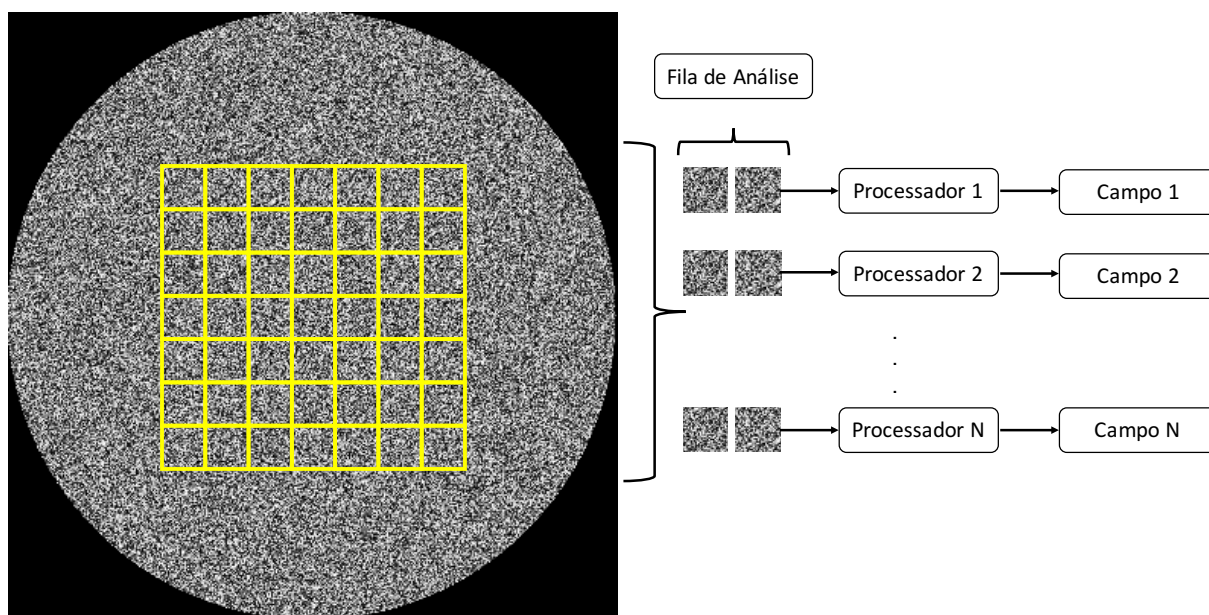
Por outro lado, técnicas de memória distribuída são utilizadas quando várias máquinas independentes fazem os devidos cálculos. Nesse caso, uma máquina é definida como mestre (*master*) e as outras são chamadas de escravos (*slaves*). O computador mestre é responsável pela distribuição de atividades entre os computadores escravos, bem como realiza a comunicação entre todas as máquinas. A comunicação entre diversas máquinas é realizada através de mensagens.

No contexto de correlação de imagens digitais, os métodos de análise com abordagem local dividem a região de interesse em diversas células menores. Nesta Seção, não serão abordadas técnicas de CAD para técnicas de abordagem global. O processo de correlação é realizado em cada uma dessas células. Dessa forma, é montada uma grade uniforme dividindo a região de interesse. A criação da grade, para avaliação de subregiões internas da região de interesse, é parte de uma técnica conhecida como "dividir para conquistar". A grade é definida pelo tamanho da célula e pelas coordenadas da fronteira da região de interesse.

Sem a aplicação de técnicas de CAD, a análise é realizada usando apenas um processador, que calcula os campos de deslocamentos para cada uma das células. A incorporação de técnicas de CAD permite uma subdivisão de tal forma que mais processadores realizem o processo de análise de várias células.

Com a grade definida, ordenam-se as células que serão enviadas para análise. Com isso, o trabalho do cálculo dos campos é realizado por vários processadores trabalhando em paralelo. A Figura 10 ilustra uma representação de processamento paralelo.

Figura 10 – Representação de processamento paralelo.



Fonte: Autor.

Para cada subregião analisada, é calculado um respectivo campo de deslocamentos. Nenhum desses campos apresenta relação com os campos calculados nas células vizinhas. Nesse caso, informações sobre continuidade entre subregiões é perdida.

2.8 Método SIFT

O método *Scale Invariant Feature Transform* (SIFT) foi proposto por Lowe (1999) como uma alternativa para reconhecimento computacional de objetos. Esse método busca nas imagens características que não variam com operações de mudança de escala, translações, rotações e transformações afins.

De acordo com Rey Otero e Delbracio (2014), o método extrai um conjunto de descritores de uma imagem através de duas operações independentes e realizadas em sequência: a detecção de pontos de interesse e a extração de um descritor associado a cada ponto. Como cada descritor independe de alguns tipos de transformações, eles são utilizados para correspondência de pontos em imagens. Rey Otero e Delbracio (2014) apresentam uma sequência de macroatividades do método SIFT para montagem do conjunto descritor dos pontos de interesse (Figura 11).

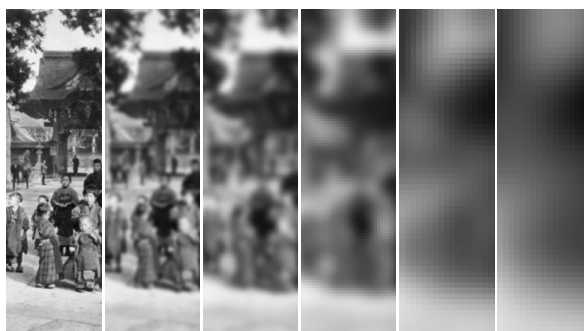
Figura 11 – Pseudo-algoritmo para SIFT.

Stage	Description
1.	Compute the Gaussian scale-space in: \mathbf{u} image out: \mathbf{v} scale-space
2.	Compute the Difference of Gaussians (DoG) in: \mathbf{v} scale-space out: \mathbf{w} DoG
3.	Find candidate keypoints (3D discrete extrema of DoG) in: \mathbf{w} DoG out: $\{(x_d, y_d, \sigma_d)\}$ list of discrete extrema (position and scale)
4.	Refine candidate keypoints location with sub-pixel precision in: \mathbf{w} DoG and $\{(x_d, y_d, \sigma_d)\}$ list of discrete extrema out: $\{(x, y, \sigma)\}$ list of interpolated extrema
5.	Filter unstable keypoints due to noise in: \mathbf{w} DoG and $\{(x, y, \sigma)\}$ out: $\{(x, y, \sigma)\}$ list of filtered keypoints
6.	Filter unstable keypoints laying on edges in: \mathbf{w} DoG and $\{(x, y, \sigma)\}$ out: $\{(x, y, \sigma)\}$ list of filtered keypoints
7.	Assign a reference orientation to each keypoint in: $(\partial_m \mathbf{v}, \partial_n \mathbf{v})$ scale-space gradient and $\{(x, y, \sigma)\}$ list of keypoints out: $\{(x, y, \sigma, \theta)\}$ list of oriented keypoints
8.	Build the keypoints descriptor in: $(\partial_m \mathbf{v}, \partial_n \mathbf{v})$ scale-space gradient and $\{(x, y, \sigma, \theta)\}$ list of keypoints out: $\{(x, y, \sigma, \theta, \mathbf{f})\}$ list of described keypoints

Fonte: Rey Otero e Delbracio (2014).

O método capta as coordenadas (x, y) dos pontos a partir de uma representação multiescalar, que consiste em um conjunto de imagens cada vez mais borradas, onde a escala σ é definida. A Figura 12 apresenta uma representação multiescalar de uma imagem. Além disso, SIFT calcula a orientação θ dominante sobre a região vizinha a cada um desses pontos. Assim, é definida uma quádrupla (x, y, σ, θ) para cada ponto de interesse.

Figura 12 – Representação multiescalar.



Fonte: Rey Otero e Delbracio (2014).

A etapa de detecção dos pontos de interesse é realizada no espaço-escala gaussiano, onde os pontos são encontrados computando os valores absolutos de um operador diferencial aplicado no espaço-escala. O operador usado é a diferença de gaussiano (*Difference of Gaussian* - DoG). Depois, a localização é refinada em nível subpixélico usando interpolação quadrática.

Em seguida, é calculada a orientação para cada ponto. Segundo Rey Otero e Delbracio (2014), essa etapa envolve três passos:

1. Somatório da distribuição local do ângulo do gradiente, normalizado com o histograma de orientação;
2. Suavização do histograma;
3. Extração de uma ou mais orientações de referência a partir do histograma suavizado.

Com as informações de escala e orientação para cada um dos pontos, monta-se o descritor do conjunto de pontos de interesse de cada imagem. Com os descritores da imagem principal e da imagem secundária, inicia-se o processo de correspondência, que é realizado comparando-se as quádruplas dos descritores.

A correspondência envolve o cálculo das distâncias entre todas as quádruplas dos pontos de interesse. Um par é considerado uma correspondência interessante quando seu valor é menor do que um determinado limite. Esse tipo de correspondência é chamada absoluta, onde o correspondente mais próximo é indicado.

Há também uma correspondência relativa, onde é calculada uma razão entre as distâncias do vizinho mais próximo e do segundo vizinho mais próximo. Essa razão também é comparada com um limite máximo.

3 AMBIENTE COMPUTACIONAL DE ALTO DESEMPENHO

3.1 Estrutura do Ambiente

O ambiente computacional de alto desempenho usando tecnologias de correlação de imagens digitais (CID) consiste em um modelo conceitual baseado em módulos e processos integrados que permitem a análise de grandezas associadas à engenharia estrutural. A organização da arquitetura do ambiente objetiva a incorporação de tecnologias para o cálculo de deslocamentos usando CID.

A arquitetura de um ambiente computacional pode ser definida como “a estrutura ou a organização de componentes de programa (módulos), a maneira através da qual esses componentes interagem e a estrutura de dados que são usadas pelos componentes” (PRESSMAN, 2009). Também, os componentes “podem ser generalizados para representar os principais elementos de um sistema e suas interações” (PRESSMAN, 2009).

Pode-se dividir o processo de análise computacional em três atividades principais. A primeira etapa corresponde ao pré-processamento, que define o objeto de estudo e o comportamento físico a ser estudado. No processo de CID, essa etapa consiste na montagem do equipamento e aquisição das imagens, juntamente com a aplicação de técnicas de processamento de imagens para eliminação de ruídos.

A segunda etapa é a análise das imagens capturadas. Nessa fase, são aplicados os principais conceitos de CID para correspondência de determinados pontos nas diferentes imagens. O campo de deslocamentos e deformações final é o produto do rastreamento desses pontos.

A terceira e última fase consiste na visualização dos resultados, com aplicação de ferramentas gráficas para organização e exibição dos dados obtidos pelo módulo de análise.

O ambiente foi idealizado seguindo esse processo, onde cada fase pode ser executada individualmente. A Figura 13 apresenta a estrutura modular do ambiente.

Figura 13 – Diagrama do modelo geral do ambiente.

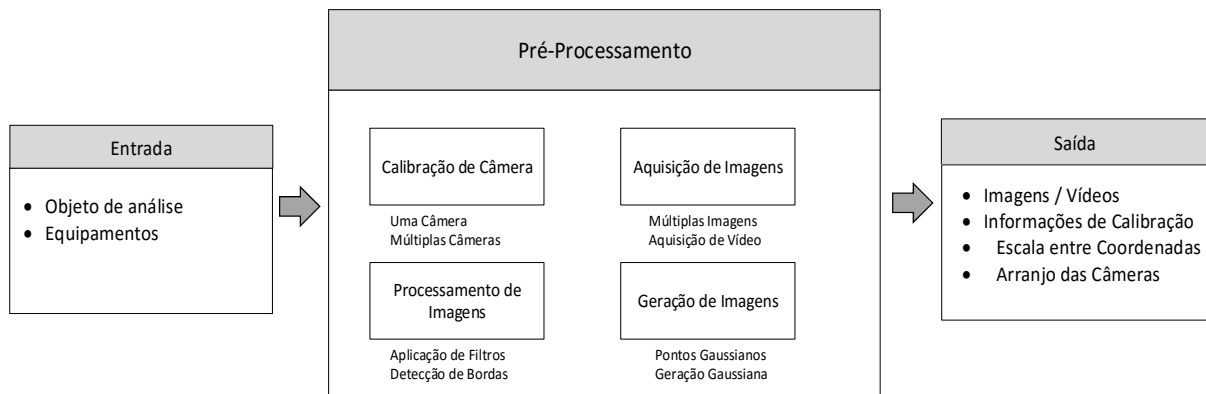


Fonte: Autor.

O módulo de pré-processamento é composto por processos de aquisição/geração de imagens, aplicação de técnicas de processamento de imagens e calibração das câmeras utilizadas. A flexibilização desse módulo permite que a aquisição de imagens leve em consideração o objeto de estudo e o comportamento físico que se deseja analisar. O módulo se encarrega de capturar e executar os filtros de imagens que anulem a maior parte dos ruídos.

Os procedimentos de calibração das câmeras, como definição de escalas entre os sistemas de coordenadas real e da imagem e disposição das câmeras, bem como técnicas para geração de imagens virtuais e aplicação de campos de deslocamentos controlados são inseridos nesse módulo. A Figura 14 mostra uma representação processual do módulo de pré-processamento. Ressalta-se que, conforme descrito na Seção Delimitação do Trabalho, o detalhamento deste módulo não é o foco do trabalho.

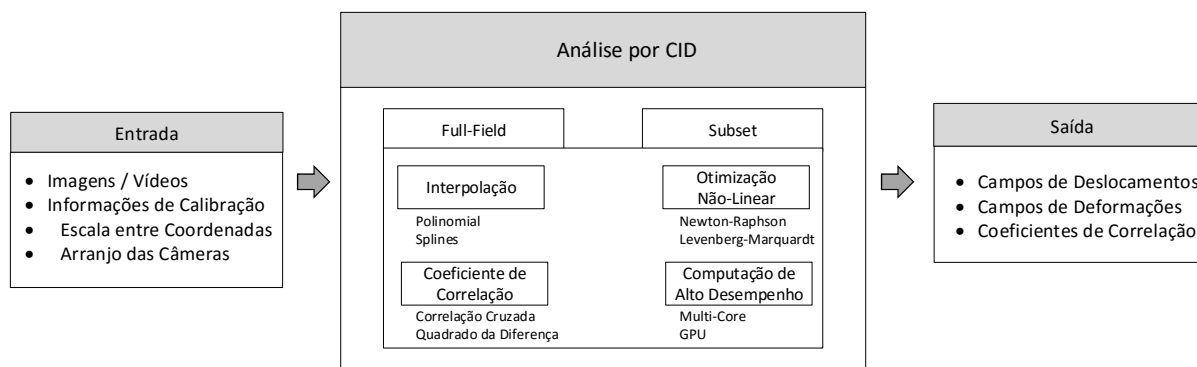
Figura 14 – Diagrama do módulo de pré-processamento.



Fonte: Autor.

O módulo de análise é composto por processos de determinação de campos de deslocamentos e imagens usando CID. Esse módulo recebe os dados do módulo de aquisição (imagens e/ou vídeo e informações de calibração da câmera) e é responsável pelo cálculo do campo de deslocamentos e deformações usando procedimentos de análise por CID. A Figura 15 apresenta um desenho esquemático desse módulo.

Figura 15 – Diagrama do módulo de análise por CID.



Fonte: Autor.

No módulo de análise são incorporadas as estratégias de análise usando CID para a região de interesse a ser avaliada. Então, mapeia-se o processo de análise e suas variantes, com identificação dos componentes de acordo com a estratégia adotada. Existem técnicas que são usadas apenas em um tipo de abordagem, assim como há componentes comuns às duas estratégias.

Dentre os componentes exclusivos, destaca-se a função campo de deslocamentos, responsável pela correspondência entre as imagens. As estratégias locais exigem um campo de deslocamentos com variáveis relativas às coordenadas do centro de cada célula. A abordagem global não apresenta essa restrição. O componente de interpolações é exclusivo da abordagem local em decorrência dessa restrição da função campo de deslocamentos.

Há, também, um conjunto de métodos comuns para as estratégias usadas. A expressão do coeficiente de correlação é um exemplo. Algumas fórmulas são mais adequadas para imagens reais, apesar de possuírem um custo computacional relativamente elevado, pois suas expressões são mais elaboradas. Outro exemplo é o processo de otimização não-linear, usado para refino dos coeficientes de ponderação aplicados às funções de campo de deslocamentos e deformações, conforme descrito anteriormente.

Por outro lado, as funções de campo de deslocamentos e deformações são diferentes de acordo com a estratégia adotada. Em estratégias locais, os campos de deslocamentos são compostos por variáveis relativas ao sistema de coordenadas local da célula analisada. Nas estratégias globais, o campo é composto por funções e variáveis relativas ao sistema global da região de interesse. Para o caso, não há aplicação de métodos de interpolação.

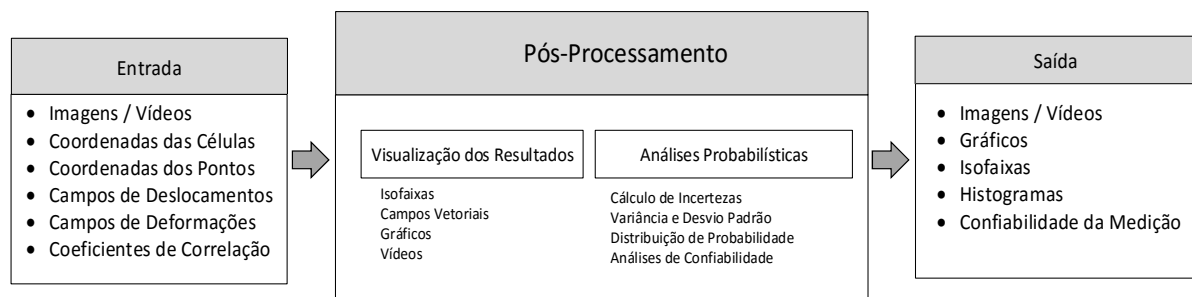
Em situações onde o campo de deslocamentos seja heterogêneo e pretende-se utilizar a estratégia local, deve-se escolher um método de interpolação mais avançado. A aplicação de uma técnica de interpolação mais elaborada, juntamente com a repetição do processo de correlação em cada subcélula, eleva o custo computacional de execução. É interessante, então, a incorporação de técnicas de computação em alto desempenho para que o desempenho da

análise não seja comprometido. Mais informações sobre os processos internos desse módulo são apresentadas em seção posterior.

O módulo de pós-processamento engloba os processos de visualização e análise de resultados. Nessa etapa, são inseridas tecnologias para apresentação dos campos de deslocamentos através de isoformas, campos vetoriais, análise de grandezas físicas, entre outras ferramentas de apresentação de resultados.

O módulo também abrange atividades de análise estatística de incertezas em relação ao modelo usado. Incorporações de técnicas estocásticas, com cálculo da influência de fatores externos (distorção das lentes e alterações na iluminação, por exemplo) e internos (escolha de funções de interpolação ou campos de deslocamentos incompatíveis, entre outros) no cálculo final do campo de deslocamentos e deformações. A Figura 16 apresenta um desenho esquemático do módulo de pós-processamento. Novamente, ressalta-se que o detalhamento deste módulo não é o foco do trabalho.

Figura 16 – Diagrama do módulo de pós-processamento.



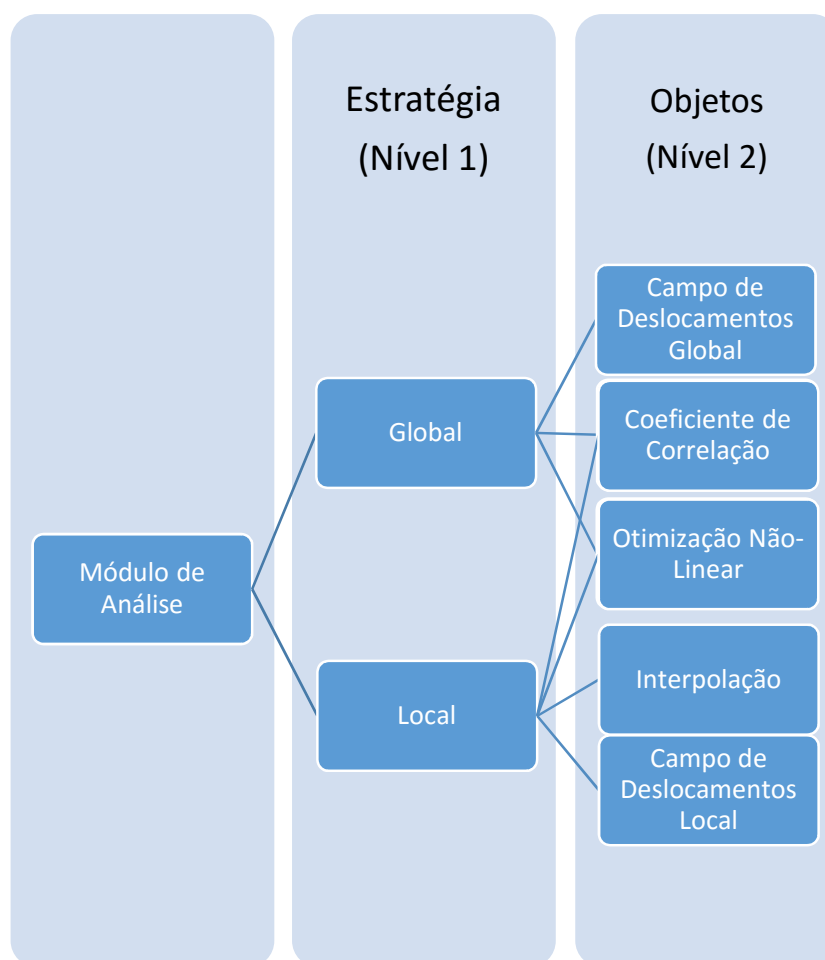
Fonte: Autor.

Conforme dito, o presente trabalho foca no módulo de análise das imagens a partir de técnicas de CID. Assim, são detalhados a seguir os componentes apresentados na proposta do ambiente computacional para análise de imagens através da CID, bem como as relações entre os procedimentos.

3.2 Componentes do Módulo de Análise

O módulo de análise é estruturado em dois módulos internos, de acordo com a abordagem adotada para a execução da análise. A abordagem local (ou *Subset*) define a utilização de um determinado subconjunto de objetos (processos), enquanto a abordagem global (*Full-Field*) usa um subconjunto diferente. Os dois subconjuntos, no entanto, utilizam objetos em comum. A Figura 17 apresenta um diagrama do módulo de análise.

Figura 17 – Estrutura interna do módulo de análise.



Fonte: Autor.

Ne Seção seguinte, são descritos os componentes internos que indicam os processos mais usados em CID, bem como suas relações entre si. Também são apresentados os processos internos para os objetos. O processo teórico da abordagem local ou global, juntamente com o mapeamento do campo de deslocamentos local e global, foi discutido na Seção 2.

3.2.1 Coeficiente de Correlação

Conforme registrado na Seção 2, o coeficiente de correlação é um indicador da semelhança entre duas células. Para o coeficiente de correlação cruzada, quanto maior o número, mais semelhantes são as regiões avaliadas. Para o coeficiente do somatório do quadrado da diferença, quanto menor o número, maior a semelhança. Como uma medida de similaridade de imagens, é importante considerar que a presença de ruídos na imagem dificulta o processo de medição.

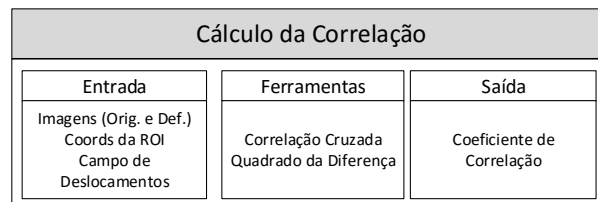
As fontes de ruídos podem ser externas, como distorções na lente ou dificuldades do sensor em captar o objeto, mudanças na iluminação ou alterações na posição do objeto de estudo ou

da câmera; ou internas, como escolha de funções de interpolação ou campos de deslocamentos incompatíveis.

As expressões mais utilizadas para o cálculo do coeficiente de correlação são as de correlação cruzada e somatório do quadrado da diferença, conforme apresentado na Seção 2. Algumas variações delas são focadas na redução de ruídos externos, calculando o coeficiente a partir de uma relação de valores médios, obtidos durante a operação.

Para o ambiente computacional, o processo de cálculo do coeficiente de correlação é independente da estratégia adotada, assim como da metodologia de minimização escolhida. Como a correlação utiliza apenas as imagens e suas coordenadas, não há adaptação do processo de cálculo. A Figura 18 apresenta o diagrama geral do processo de cálculo do coeficiente de correlação.

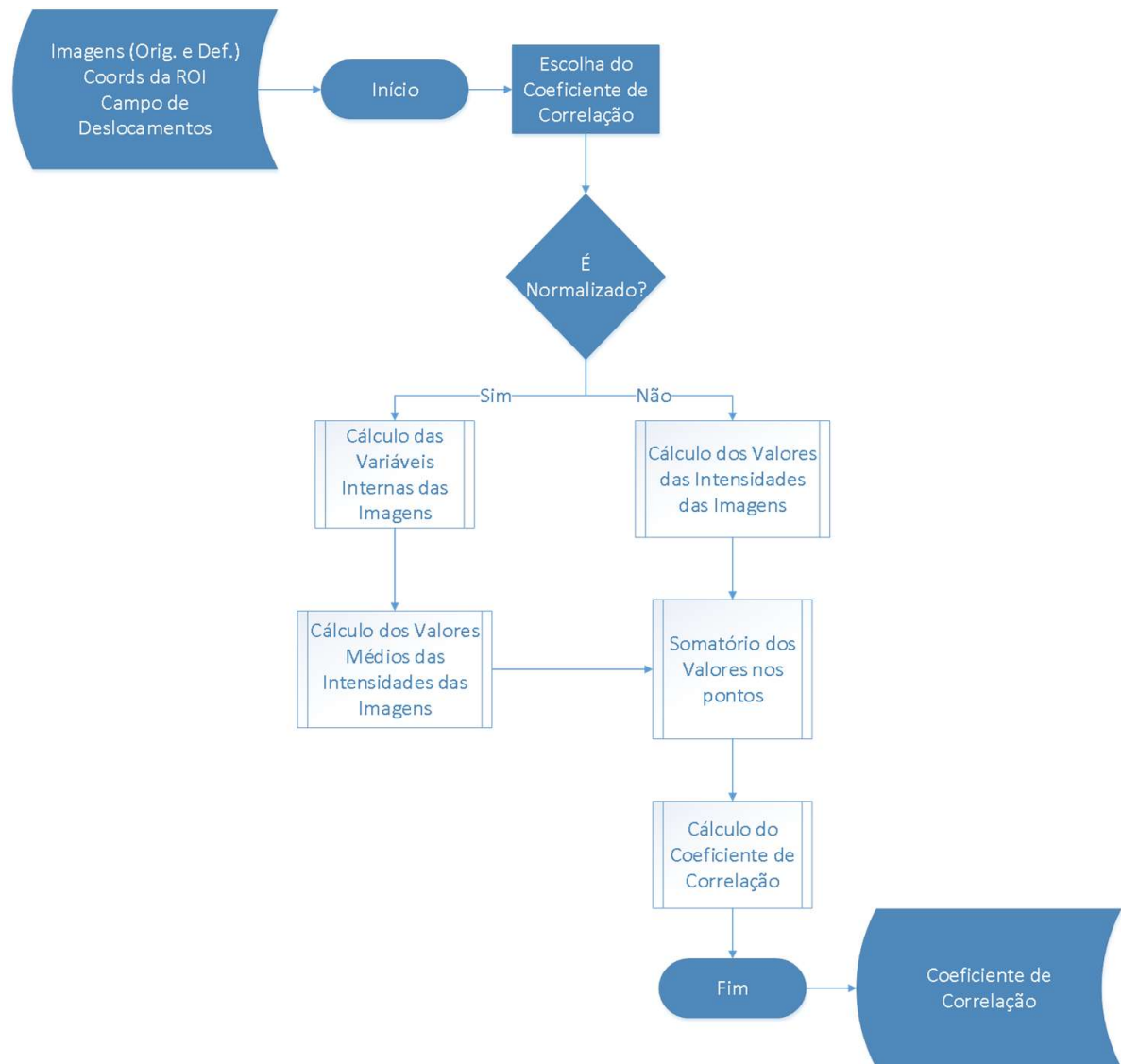
Figura 18 – Fases do processo de cálculo do coeficiente de correlação.



Fonte: Autor.

É interessante observar que os valores de entrada necessários à execução do cálculo são definidos diretamente pelos processos de estratégia e otimização adotados. Assim, a escolha do procedimento para cálculo pode ser realizada independente dos outros processos. A Figura 19 ilustra a execução do processo de cálculo dos coeficientes de correlação.

Figura 19 – Diagrama simplificado do cálculo do coeficiente de correlação.



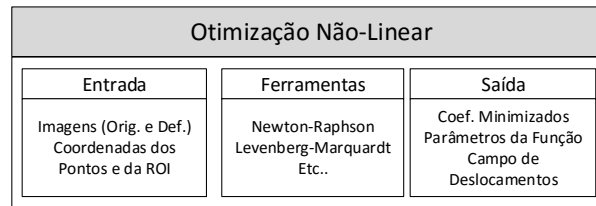
Fonte: Autor.

No processo acima, as variáveis internas se referem aos valores de f_m , \bar{F} e ΔF , conforme descrito nas Equações 2.3 e 2.5.

3.2.2 Otimização Não-Linear

Os processos de otimização não-linear são comuns aos dois tipos de abordagens previamente discutidos. Os métodos de otimização estimam os valores ótimos para os coeficientes de ponderação da função campo de deslocamentos. Esses valores ocorrem quando a expressão do coeficiente de correlação retorna o menor valor possível. Dentro do contexto da CID, as técnicas de otimização mais comuns são as de Newton-Raphson (NR) e Levenberg-Marquardt (LM). A Figura 20 ilustra as fases do processo de otimização.

Figura 20 – Fases do processo de otimização não-linear.

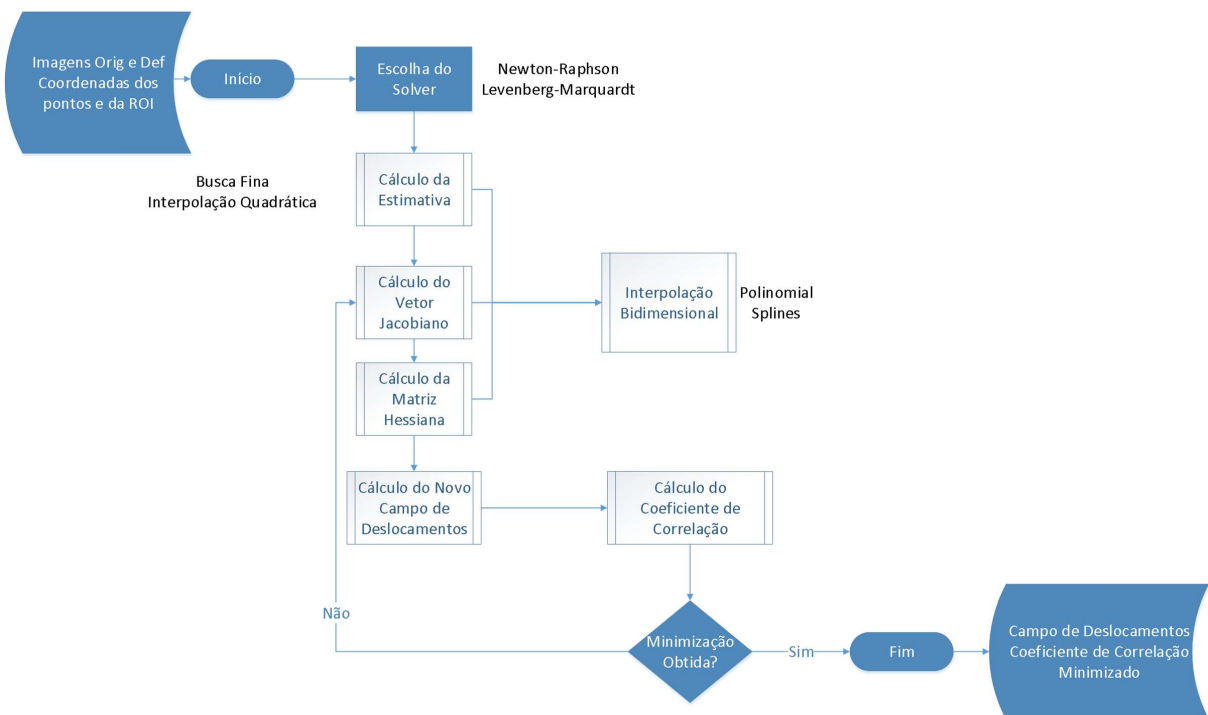


Fonte: Autor.

A minimização do coeficiente engloba o cálculo do coeficiente de correlação, correspondência de imagens através do campo de deslocamentos e, caso a abordagem de análise seja local, atividades de interpolação subpixélica. Seu papel integra os outros componentes necessários à execução do cálculo do comportamento deformacional usando CID.

Os métodos NR e LM utilizam os valores das derivadas da função resíduo (conforme Equação 2.6) para realizar o processo de otimização. O processo inicia com a definição de uma estimativa inicial para os parâmetros, seguida do refino da solução. Executa-se um cálculo iterativo e incremental que melhora o cálculo da estimativa, buscando pelos valores mínimos da função bidimensional do coeficiente de correlação. A Figura 21 apresenta o fluxo de otimização não-linear, com exemplos de possíveis métodos que podem ser incorporados em partes do processo.

Figura 21 – Diagrama simplificado com processo de otimização não-linear - abordagem local de análise.



Fonte: Autor.

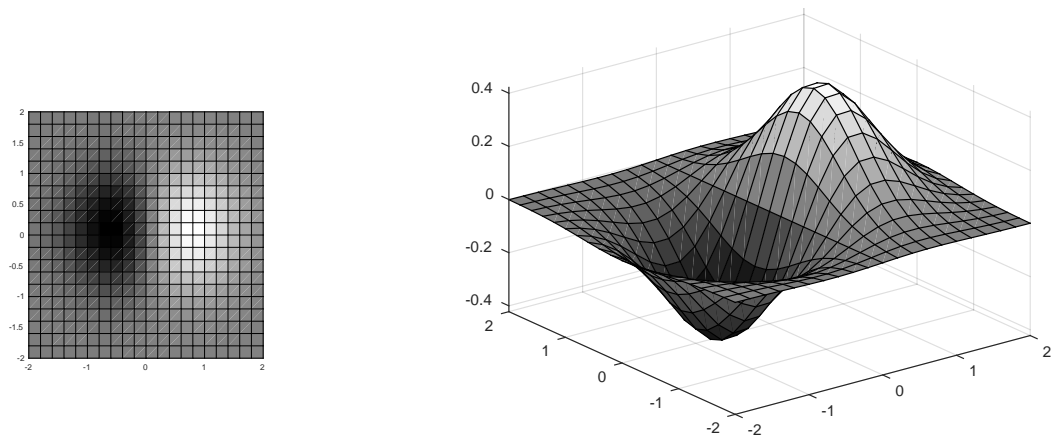
Nesse processo, também há escolha dos métodos para cálculo da estimativa inicial, da interpolação bidimensional e do coeficiente de correlação de acordo com o tipo de ensaio realizado e com as grandezas que se procura avaliar.

O fluxo anterior serve para os métodos de otimização já descritos (NR e LM) e quaisquer outros métodos que envolvam o cálculo do vetor gradiente e da matriz hessiana. No entanto, outros métodos de otimização não-linear podem ser adaptados para aplicação em CID no modelo descrito desde que sejam mantidos os produtos de entrada e saída acima descritos.

3.2.3 Interpolação

A interpolação apresenta um papel importante no cálculo dos campos de deslocamentos por CID dentro da abordagem local. Sua introdução permite a conversão de uma imagem digital (que apresenta uma quantidade limitada e discreta de informações) em uma função bidimensional contínua. A Figura 22 ilustra a conversão entre uma imagem e sua respectiva função.

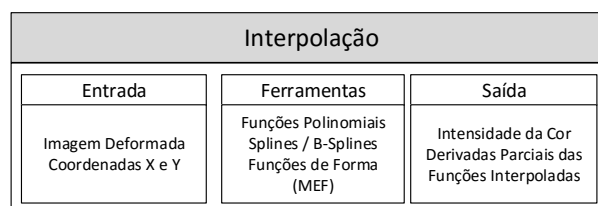
Figura 22 – Conversão entre imagem e função contínua.



Fonte: Autor

Dessa forma, determinam-se as intensidades dos tons de cinza da imagem em posições subpixélicas. A Figura 23 apresenta as principais fases de interpolação dentro da arquitetura.

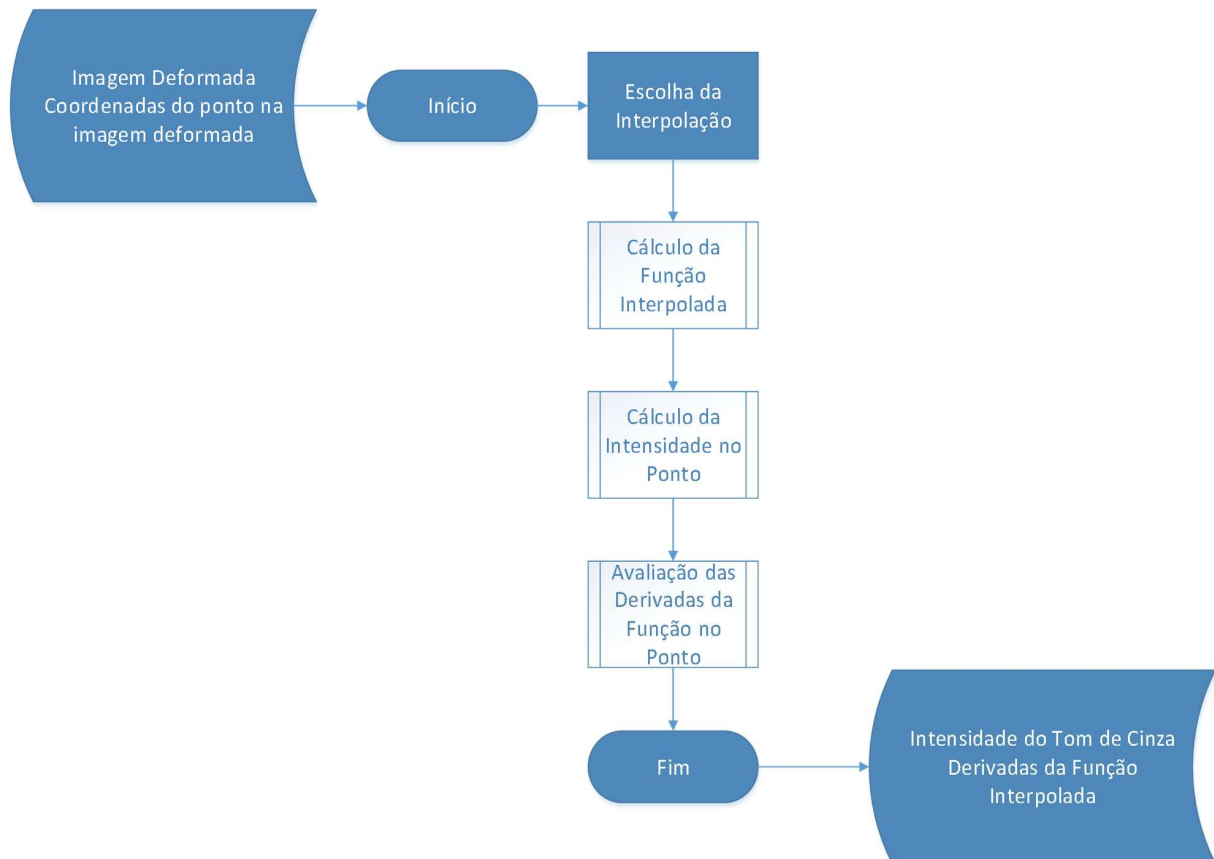
Figura 23 – Fases do processo de otimização não-linear.



Fonte: Autor.

No processo, é importante que a interpolação retorne as derivadas parciais em relação às coordenadas definidas juntamente com o tom de cinza entre os *pixels* da imagem. Com esses valores, consegue-se determinar os valores do vetor gradiente e da matriz Hessiana durante o processo de otimização não-linear. A Figura 24 ilustra o processo simplificado para interpolação no ambiente computacional.

Figura 24 – Diagrama simplificado do processo de interpolação.



Fonte: Autor.

O cálculo das funções de interpolação ocorre de acordo com a função escolhida para definir o comportamento dos tons de cinza da imagem deformada. A partir desse cálculo, avalia-se a intensidade do tom de cinza no ponto definido pela função campo de deformações, que é a primeira derivada da função campo de deslocamentos. O cálculo das derivadas pode ser realizado analiticamente ou utilizando o Método das Diferenças Finitas de acordo com o método usado para encontrar a função.

Na literatura, são apresentadas diversas maneiras diferentes de se calcular o valor da intensidade do tom de cinza em coordenadas entre *pixels*. Schreier, Braasch e Sutton (2000) apresentam interpolações bilineares, bicúbicas polinomiais e splines bicúbicas e biquínticas em diversas aplicações na literatura, bem como recomenda a utilização de splines bicúbicas e biquínticas.

3.3 Computação de Alto Desempenho

As técnicas de computação de alto desempenho são incorporadas pelas vantagens que apresentam quando o cálculo da CID é realizado com imagens de alta resolução e dentro de uma divisão celular onde a quantidade de janelas é muito grande.

A aplicação do método é inicialmente prevista para a abordagem *Subset*, onde o cálculo do campo de deslocamentos é realizado de maneira independente para cada célula. Dessa forma, executa-se o procedimento de otimização não-linear usando sistemas de processamento paralelo em cada uma das células.

Dependendo do tipo de *hardware* utilizado (processador *multi-core* ou GPU), o processamento em paralelo utiliza tecnologias de memória compartilhada ou distribuída. A Figura 25 ilustra as fases do processo de computação em alto desempenho.

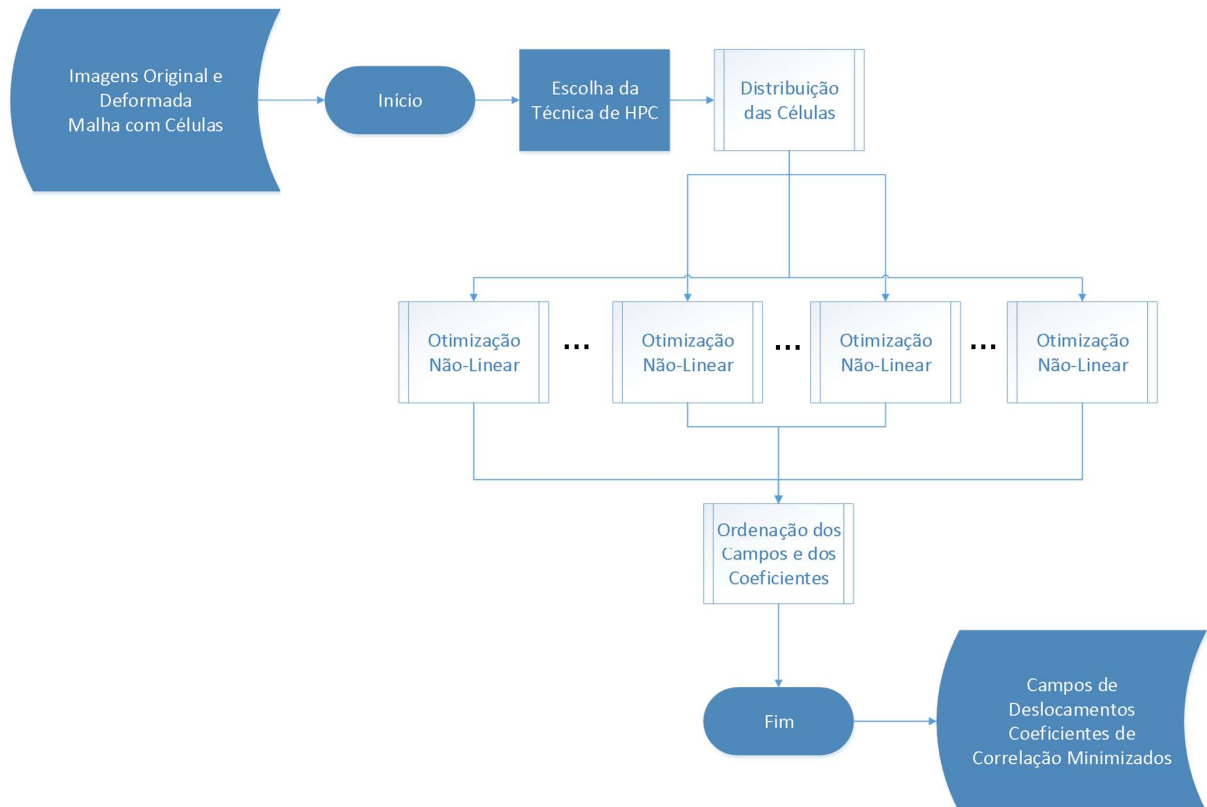
Figura 25 – Fases do processo de computação de alto desempenho.



Fonte: Autor.

Os campos de deslocamentos na fase de saída se referem aos parâmetros encontrados no processo de otimização para cada célula. Dessa forma, os produtos de saída dessa fase consistem em uma matriz com os parâmetros dos campos de deslocamentos e um vetor com os coeficientes de correlação minimizados. Tanto a matriz quanto o vetor devem estar ordenados de acordo com a orientação da malha passada como entrada. A Figura 26 apresenta o fluxograma simplificado da aplicação de técnicas de computação de alto desempenho.

Figura 26 – Diagrama simplificado para Computação de Alto Desempenho.



Fonte: Autor.

A distribuição dos processos de otimização não-linear, bem como a ordenação dos campos de deslocamentos e coeficientes de correlação calculados, é definida de acordo com o tipo de técnica de computação em alto desempenho utilizada. A incorporação da técnica deve levar em consideração o tipo de *hardware* utilizado e, a partir daí, verificar as possibilidades de inserção de estratégias para utilização de métodos com memória compartilhada ou distribuída.

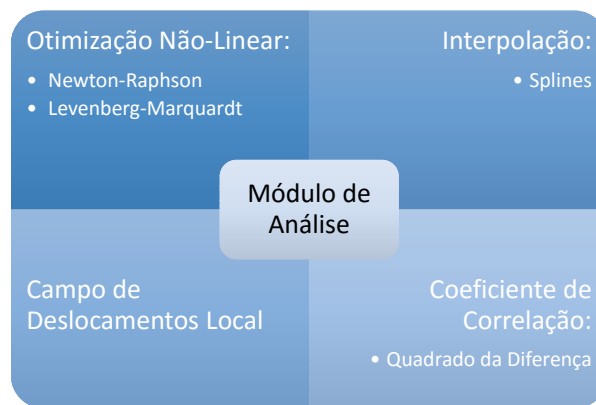
4 PROTÓTIPO DO AMBIENTE

O protótipo é a reprodução da arquitetura do ambiente computacional proposto. O desenvolvimento do protótipo representa um aspecto prático da arquitetura do ambiente computacional de alto desempenho. Sua implementação serve para validar a arquitetura apresentada, bem como a execução do procedimento de cálculo dos campos de deslocamentos através da otimização dos coeficientes de correlação.

Como já afirmado, apenas o desenvolvimento do módulo de análise será realizado. O presente trabalho torna como base para alguns componentes o algoritmo desenvolvido por Ferraz (2013) para implementar outras técnicas de otimização não-linear e interpolação em graus mais elevados.

Utiliza-se como ambiente para desenvolvimento o *Matrix Laboratory* (Matlab) (The MathWorks Inc., 2015). Assim, desenvolve-se uma nova revisão incluindo avanços computacionais para reduzir os problemas encontrados. A Figura 27 apresenta a composição do módulo de análise e dos componentes desenvolvidos.

Figura 27 – Adaptações realizadas ao fluxo de Ferraz (2013).



Fonte: Autor.

Os componentes descritos no diagrama acima são os métodos implementados no protótipo para análise por correlação de imagens digitais. As seções seguintes apresentam os métodos incorporados em cada uma das partes do ambiente computacional. Cada seção está delimitada pelo tipo de técnica utilizada.

A comunicação entre componentes é realizada diretamente através da memória RAM, uma vez que as informações usadas por cada componente são calculadas diretamente durante o processo de análise. Dessa forma, não há necessidade de escrita de nenhuma informação em arquivo de dados.

É importante ressaltar que, por ser um trabalho em desenvolvimento, parte das ativida-

des planejadas ainda não foi executada. Nesse caso, apesar de o ambiente computacional envolver técnicas de computação de alto desempenho, estas técnicas ainda não foram incorporadas ao protótipo.

4.1 Tipo de Abordagem e Campo de Deslocamentos

A estratégia adotada para a análise das imagens digitais no protótipo é de abordagem local. Com isso, a função campo de deslocamentos apresenta como variáveis as distâncias relativas às coordenadas do centro da célula local. Atualmente, o protótipo realiza apenas análises de imagens digitais usando abordagem local, conforme usado por Ferraz (2013).

Para o protótipo, adota-se uma função campo de deslocamentos com variáveis do segundo grau, conforme descrito pela Equação 4.1. Esse tipo de função também foi usada por Ferraz (2013).

$$x_d = x_0 + p_0 + p_2\Delta x + p_4\Delta y + \frac{1}{2}p_6(\Delta x)^2 + \frac{1}{2}p_8(\Delta y)^2 + p_{10}\Delta x\Delta y \quad (4.1a)$$

$$y_d = y_0 + p_1 + p_3\Delta x + p_5\Delta y + \frac{1}{2}p_7(\Delta x)^2 + \frac{1}{2}p_9(\Delta y)^2 + p_{11}\Delta x\Delta y \quad (4.1b)$$

onde x_d e y_d representa as coordenadas da imagem deformada e x_0 e y_0 as coordenadas da imagem original. Os valores de Δx e Δy são distâncias nos eixos X e Y entre uma posição qualquer e a posição central da célula.

Por se tratar de um campo de deslocamentos com presença de variáveis do segundo grau e termos cruzados, não se viu necessidade de incorporar outro tipo de campo de deslocamentos. Da mesma forma, como o protótipo não executa análises globais em imagens, não foi incorporado nenhum campo de deslocamentos que pudesse ser usado nesse tipo de análise.

4.2 Coeficiente de Correlação

O cálculo do coeficiente de correlação, apesar de ser um componente independente no processo, possui papel fundamental na análise. Seu valor é minimizado de acordo com a otimização executada.

A expressão do coeficiente de correlação usada por Ferraz (2013) é uma variação do somatório do quadrado da diferença (SSD). A Equação 4.2 apresenta a expressão original (reescrita da Equação 2.6) e a Equação 4.3 mostra a expressão usada por Ferraz (2013).

$$C_{SSD} = \sum_{i=-M}^M \sum_{j=-M}^M [F(x_i; y_j) - G(\tilde{x}_i; \tilde{y}_j)]^2 \quad (4.2)$$

$$C_{SSD} = \frac{\sum_{i=-M}^M \sum_{j=-M}^M [F(x_i; y_i) - G(\tilde{x}_i; \tilde{y}_j)]^2}{\sum_{i=-M}^M \sum_{j=-M}^M [F(x_i; y_i)]^2} \quad (4.3)$$

As expressões para cálculo do coeficiente de correlação incorporadas ao protótipo consistem no somatório do quadrado da diferença simples (Equação 4.2) e a variação normalizada em zero do somatório do quadrado da diferença (*Zero Normalized Sum of Squared Difference-ZNSSD*), apresentada na Equação 4.4 e também descrita na Equação 2.7b.

$$C_{ZNSSD} = \sum_{i=-M}^M \sum_{j=-M}^M \left[\frac{F(x_i; y_j) - f_m}{\Delta F} - \frac{G(\tilde{x}_i; \tilde{y}_j) - g_m}{\Delta G} \right]^2 \quad (4.4)$$

Nessa equação, o cálculo do ZNSSD é realizado a partir da avaliação dos valores de f_m , g_m , ΔF e ΔG , que estão apresentados na Equação 2.5. Quando introduzidos na expressão, eliminam-se alguns ruídos existentes na imagens. Os valores acabam por serem normalizados por valores médios das intensidades das imagens original e deformada.

4.3 Interpolação

A interpolação adotada por Ferraz (2013) para avaliação das intensidades em posições entre *pixels* é bilinear para a estimativa inicial e *splines* bicúbicas para o refinamento do código. A técnica bicúbica utilizada por ele é a *spline* bidimensional de grau 3, onde se utiliza uma célula 3x3 ao redor do ponto subpixélico. O processo de interpolação foi descrito na Seção 2.

O método, da forma como foi descrito por Ferraz (2013), teve seu procedimento extrapolado para que se obtivessem funções de maior grau, como 5, 7 e 9, que podem representar melhores aproximações. O procedimento de interpolação está descrito em Ferraz (2013) para o caso bicúbico. Aqui, será descrito para o caso biquíntico (grau 5), que é recomendado por Schreier, Braasch e Sutton (2000).

As interpolações de *spline* apresentam como uma de suas principais características a possibilidade de se incluir informações de continuidade para as derivadas da função. Pode-se, então, utilizar as Equações 4.5 para realizar os cálculos. As equações, reescritas para o caso de funções no grau 5, estão apresentadas a seguir.

$$g(h; t) = \sum_{m=0}^5 \sum_{n=0}^5 \alpha_{mn} h^m t^n \quad (4.5a)$$

$$g_x = g_h(h, t) = \sum_{m=0}^5 \sum_{n=0}^5 m \alpha_{mn} h^{m-1} t^n \quad (4.5b)$$

$$g_y = g_t(h, t) = \sum_{m=0}^5 \sum_{n=0}^5 n \alpha_{mn} h^m t^{n-1} \quad (4.5c)$$

$$g_{xy} = g_{ht}(h, t) = \sum_{m=0}^5 \sum_{n=0}^5 mn \alpha_{mn} h^{m-1} t^{n-1} \quad (4.5d)$$

Para o caso biquíntico, os coeficientes α_{mn} a serem calculados são 36. A célula usada para o caso é 3×3 *pixels*. Com isso, são utilizados 9 *pixels*, onde cada *pixel* contribui com 4 informações (intensidade, derivadas em cada um dos eixos e derivada cruzada). Dessa forma, obtêm-se 36 informações, o que é suficiente para construir o sistema de equações lineares.

A resolução do sistema passa pela inversão da matriz de coeficientes, que é realizada previamente, uma vez que esse cálculo não é alterado pelos valores das intensidades dos *pixels*. As alterações acontecem no vetor das constantes, que é composto pelos valores das intensidades e das derivadas calculadas através do método das diferenças finitas. A Equação

$$[A]_{36 \times 36} \{\alpha_{mn}\}_{36 \times 1} = \begin{pmatrix} f^1 \\ \dots \\ f_x^1 \\ \dots \\ f_y^1 \\ \dots \\ f_{xy}^1 \\ \dots \end{pmatrix}_{36 \times 1} \quad (4.6)$$

ilustra o sistema de equações.

A resolução do sistema resulta nos valores dos coeficientes que definem a *spline* de grau 5. De maneira análoga, são desenvolvidas implementações que interpolam funções bidimensionais de graus 7 e 9. As células necessárias são da ordem 8 e 10, respectivamente, e suas matrizes são de ordem 64 e 100.

4.4 Otimização Não-Linear: Métodos de Newton-Raphson e Levenberg-Marquardt

4.4.1 Estimativa Inicial

O cálculo da estimativa inicial implementado por Ferraz (2013) é o mesmo utilizado em Sutton et al. (1986). O processo de busca fina é realizado através de força bruta, ao se calcular

os coeficientes de correlação para campos de deslocamento com variações dentro de uma janela 4x4. As variações realizadas são feitas com números inteiros. Em ambos os trabalhos, utilizou-se a interpolação bilinear para calcular os valores nessa primeira etapa do processo.

As alterações realizadas nessa etapa do algoritmo consistem no aumento da janela de busca (5x5), que amplia a abrangência da varredura, a troca para a interpolação por *spline* bicúbica e a implementação de um algoritmo de refino da estimativa inicial, utilizado após a busca fina.

O algoritmo de refino, conhecido como algoritmo de pico (*Peak Algorithm*) e descrito em Chen et al. (1993), consiste em uma interpolação bidimensional de segundo grau, realizada em uma célula 3x3 ao redor do ponto onde o campo de deslocamentos apresenta o menor coeficiente de correlação. A interpolação permite que o campo de deslocamentos possa ter valores subpixélicos sem que seja necessário fazer testes de força bruta com intervalos menores do que um inteiro.

4.4.2 Refinamento da Solução

Em conjunto com a estimativa inicial, o processo do refinamento do campo é a principal etapa do cálculo do campo de deslocamentos e deformações. A partir da estimativa, o processo de análise da imagem é iniciado, de maneira a encontrar os valores representativos dos campos de deslocamento. É importante que a estimativa esteja relativamente próxima do campo real.

Ferraz (2013) utilizava para a otimização o algoritmo de Newton-Raphson, onde se calculava o vetor gradiente dos pontos e uma aproximação da segunda derivada (matriz hessiana) a partir da suposição de que o valor da estimativa inicial está bem próximo do campo real. O produto da inversa da matriz hessiana pelo vetor gradiente resulta no incremento a ser somado ao valor da iteração atual e esse processo segue iterativamente até que algum critério de convergência seja atingido. A Equação 4.7 ilustra a expressão usada.

$$\mathbf{H}(f(x_k))\Delta\mathbf{x}_{k+1} = -\mathbf{J}(f(x_k)) \quad (4.7)$$

Na equação, f é a função do coeficiente de correlação adotado, $\mathbf{H}(f)$ é a matriz hessiana, $\mathbf{J}(f)$ é o vetor jacobiano e $\Delta\mathbf{x}_{k+1}$ é o incremento a ser somado no campo de deslocamentos da iteração atual.

Visando melhorar a velocidade de convergência do método, adota-se a implementação do processo de Levenberg-Marquardt, como descrito por Madsen, Nielsen e Tingleff (2004). O procedimento de Madsen, Nielsen e Tingleff (2004) é usado para o ajuste de uma função tendo por base o Método dos Mínimos Quadrados, que se encaixa na expressão de correlação por SSD.

O cálculo da matriz hessiana pode ser bastante complexo com o aumento da célula. Os métodos do tipo Quasi-Newton, como o método de Levenberg-Marquardt, utilizam uma

aproximação da hessiana para contornar o problema de minimização de resíduos. A matriz hessiana é calculada como

$$H_{NXN} = J_{NXM}^T J_{MXN}. \quad (4.8)$$

Apesar de o cálculo da matriz hessiana ser mais simples na forma apresentada, ainda não há garantia de que sua inversa, necessária para a resolução do sistema, exista. Para evitar que a hessiana seja singular, há a introdução de um coeficiente de amortecimento μ que é multiplicado pela matriz identidade e somado ao produto da matriz jacobiana. Assim, a expressão fica

$$H_{NXN} = J_{NXM}^T J_{MXN} + \mu I_{NXN}. \quad (4.9)$$

O parâmetro μ é recalculado para cada nova iteração. A introdução de μ permite que o método apresente comportamentos diferentes, de acordo com a iteração realizada. Toda vez que $\mu > 0$, a matriz hessiana é positiva definida, o que indica que o incremento está em uma direção de descida; quando o valor de μ for grande, o valor do incremento será pequeno, mas na direção máxima de descida.

O valor inicial de μ é proporcional ao maior valor encontrado na matriz, que é obtida ao se multiplicar os jacobianos. A razão de proporção é definida pelo usuário e, nesse caso, opta-se pela unidade. Madsen, Nielsen e Tingleff (2004) altera o parâmetro de amortecimento μ de acordo com um outro parâmetro denominado razão de ganho (*gain ratio*), que é definido como

$$\rho = \frac{f(x_k) - f(x_{k+1})}{L(0) - L(x_i)}. \quad (4.10)$$

Nessa equação, $L(0) - L(x_i)$ indica o ganho obtido através de um modelo linear, descrito como

$$L(0) - L(x_i) = \frac{1}{2} x_k^T (\mu x_k - J(f(x_k))). \quad (4.11)$$

Na prática, quando o valor de ρ for grande, pode-se diminuir o valor de μ de modo que o próximo passo esteja mais próximo da convergência. Quando ρ for pequeno, a aproximação de $L(x_k)$ é ruim e, por isso, deve-se aumentar o valor de μ para buscar a direção máxima de descida.

Segue apresentado o pseudo-algoritmo (Figura 28) para o Método de Levenberg-Marquardt como apresentado por Madsen, Nielsen e Tingleff (2004).

Figura 28 – Pseudo-Algoritmo do Método de Levenberg-Marquardt.

Método de Levenberg-Marquardt
<pre> k = 0; v = 2; x = x₀; A = J(x)^TJ(x); g = J(x)^Tf(x); found = (g _∞) ≤ ε₁; μ = τ · max{a_{ii}}; while (not found) and (k < k_{max}) { k=k+1; Solve (A + μI)h_{lm} = -g ; if h_{lm} ≤ ε₂(x + ε₂) found = true else { x_{new} = x + h_{lm} ρ = (F(x) - F(x_{new}))/(L(0) - L(h_{lm})) if ρ > 0 { x = x_{new} A = J(x)^TJ(x); g = J(x)^Tf(x); found = (g _∞) ≤ ε₁; μ = μ max{1/3, 1 - (2ρ - 1)³}; v = 2; } else μ = μv; v = 2v; } } </pre>

Fonte: Gattass (2008).

5 RESULTADOS

Os estudos de casos são usados para validação do protótipo e da arquitetura desenvolvidos. Optou-se pelo uso de modelos de transformação de imagens baseados em funções de deslocamentos analíticas encontradas na literatura de Engenharia Civil. A aplicação de estudos de casos com soluções analíticas permite a comparação dos resultados calculados pelo protótipo com os valores de referência, calculados do modelo de transformação. Dessa forma, consegue-se delimitar as margens de erro do cálculo usando CID a partir dos comportamentos analisados nos estudos de caso.

Além de avaliar as incertezas apresentadas pelo protótipo, realiza-se uma comparação entre o ambiente computacional desenvolvido no presente trabalho e o algoritmo desenvolvido no trabalho de Ferraz (2013). O objetivo é verificar se as tecnologias incorporadas ao protótipo representam avanços na performance, quando comparado com o algoritmo desenvolvido por Ferraz (2013).

Os exemplos são analisados utilizando o algoritmo inicial, conforme implementado por Ferraz (2013), e o ambiente computacional de alto desempenho em seu estado atual, com as modificações realizadas. Os estudos de casos usados para comparação são os mesmos apresentados por Ferraz (2013) em sua dissertação, com arquivos gerados computacionalmente.

Os resultados apresentados são avaliados em dois momentos. Inicialmente, são comparados com os valores analíticos, obtidos através do modelo usado na geração das imagens. Na sequência, são comparados com os resultados obtidos pelo algoritmo proposto por Ferraz (2013).

Para os exemplos 1 a 3, onde as imagens têm 101×101 *pixels*, são realizadas análises no centro das imagens, com tamanhos de células que variam de 10 a 60 *pixels*, aumentando de 5 em 5 *pixels*. Para os exemplos 4 e 5, onde as imagens são maiores (401×401 *pixels*) e de 10 a 370 *pixels*, aumentando de 20 em 20 *pixels*. Para ambas as análises, são inseridas como critérios de tolerância uma convergência mínima de 10^{-9} *pixels* entre os valores das iterações ou o máximo de 300 iterações realizadas.

As comparações realizadas entre os valores calculados pelo algoritmo e os valores reais estão definidas pelos erros absolutos, medidos em *pixels*. As comparações entre os algoritmos de cálculo do campo de deslocamentos são restritas ao tempo de processamento usado, número de iterações executadas e coeficiente de correlação final calculado por cada ambiente para variados tamanhos de janelas.

Para o protótipo do ambiente computacional atual, utiliza-se a interpolação bidimensional de *splines* de grau 7, juntamente com o método de Levenberg-Marquardt. O algoritmo de Ferraz (2013) usa Newton-Raphson e, em casos onde a matriz hessiana calculada é singu-

lar, é inserida uma alteração onde para que o método calcule a matriz pseudo-inversa, que é usada na determinação do incremento. O método de interpolação usado por Ferraz (2013) é a interpolação de *splines* bicúbicas.

Os dois algoritmos são executados em um computador com processador Intel Core i5-4570 de 3.20 GHz e 16 GBs de memória RAM com Windows 10 instalado.

5.1 Geração das Imagens

As imagens usadas nas análises dos exemplos tratados são geradas computacionalmente. A geração das imagens originais são realizadas usando um algoritmo randômico que determina as intensidades dos tons de cinza, onde o intervalo de variação é de 0 a 255, para cada *pixel*.

Para a geração da imagem deformada, são aplicados modelos de transformação nas imagens originais. Cada modelo de transformação é baseado em comportamentos conhecidos, de solução analítica, encontrados na literatura de Engenharia Estrutural, a saber: translação de corpo rígido, tração uniaxial, flexão pura, compressão triaxial e compressão diametral. Os exemplos serão melhor descritos nas seções seguintes.

Geralmente, os modelos de transformação calculavam posições entre *pixels* para os pontos das imagens originais. Então, aplica-se um método de interpolação para calcular as intensidades nas posições inteiras da imagem. Usa-se a função *ScatteredInterpolant* do Matlab (The MathWorks Inc., 2015), que realiza interpolações bidimensionais em conjuntos de dados esparsos. Os parâmetros usados são as posições X e Y , a intensidade dos *pixels* e o método usado para interpolação, onde se escolhe a interpolação natural por interpolar uma superfície de continuidade C^1 .

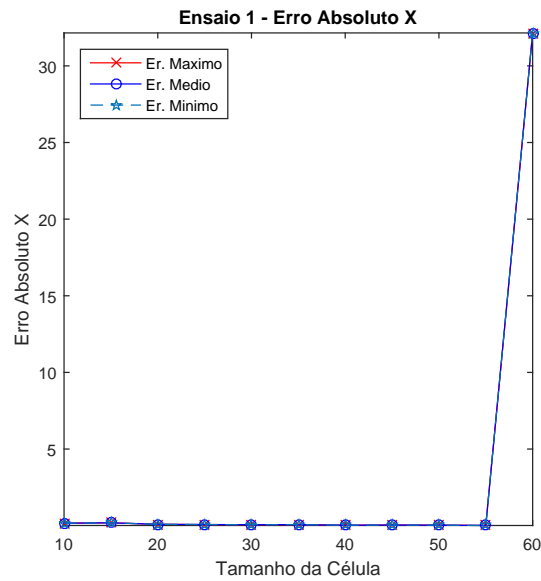
A interpolação usada gera um turvamento da imagem pelas operações realizadas, que usam *pixels* vizinhos. Apesar de o valor ser mais preciso do que no uso de uma interpolação bilinear ou *nearest neighbor*, o turvamento não é encontrado em imagens reais. Na realidade, as imagens captadas de um elemento sofrem alterações nas posições e também nas intensidades dos *pixels*, o que dificulta sua correlação.

5.2 Validação do Protótipo

5.2.1 Exemplo 1

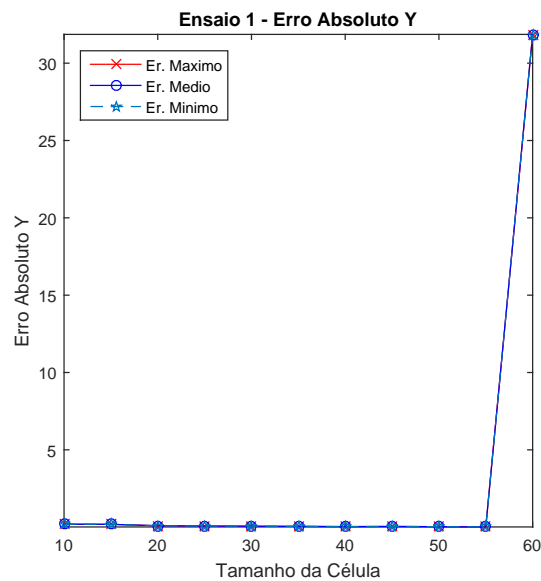
O primeiro exemplo consiste em uma translação de corpo rígido, que implica uma movimentação total da imagem por determinados valores na horizontal e na vertical. A Figura 29 apresenta a imagem original e a imagem deformada e a Tabela 1 indica o campo de deslocamentos utilizado para a reconstrução da imagem. Para efeitos comparativos, a escala global é de 1mm/*pixel* nos exemplos 1 a 3.

Figura 30 – Erro absoluto na direção X – Exemplo 1.



Fonte: Autor.

Figura 31 – Erro absoluto na direção Y – Exemplo 1.



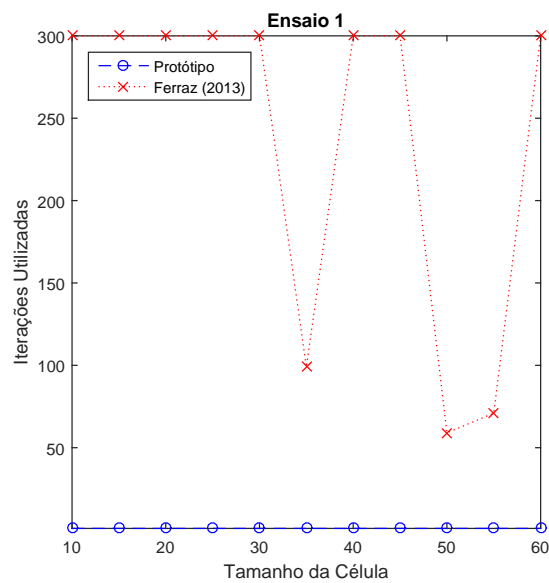
Fonte: Autor.

Pode-se verificar que, em ambos os casos, os erros máximos se mantiveram abaixo de 1 *pixel* de variação, enquanto os erros mínimos são nulos até o cálculo com célula de tamanho 60x60. Em termos de escala, 1 *pixel* equivale a 1 mm. Os erros mais extremos acontecem nas regiões de fronteira da célula, onde as coordenadas se distanciam do centro. Quanto maior a célula, maior o erro embutido. Percebe-se também que os resultados apresentam divergências em células de tamanho 60x60, quando a proximidade com a fronteira é maior.

Na comparação entre o protótipo e o algoritmo de Ferraz (2013), são coletados dados para o número de iterações, tempo de processamento em segundos e coeficiente de correlação final obtido. As Figuras 32, 33 e 34 mostram os valores calculados.

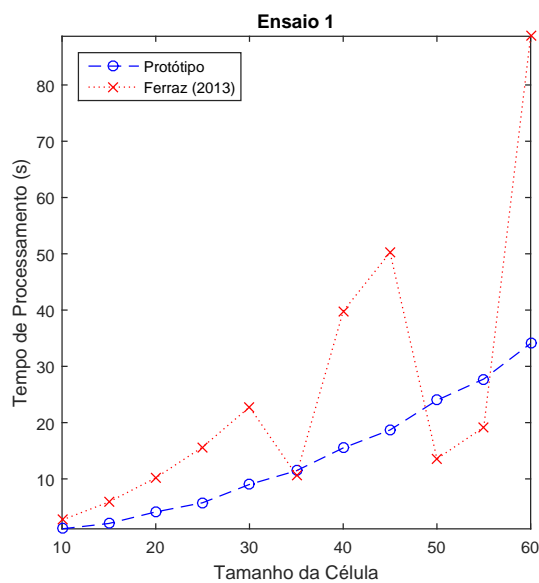
Em relação ao coeficiente de correlação, quanto menor seu valor, mais semelhantes são as regiões de interesse. Da mesma forma, quanto menor o tempo de processamento, bem como a quantidade de iterações realizadas, melhor é o desempenho do ambiente computacional.

Figura 32 – Número de iterações por tamanho de célula – Exemplo 1.



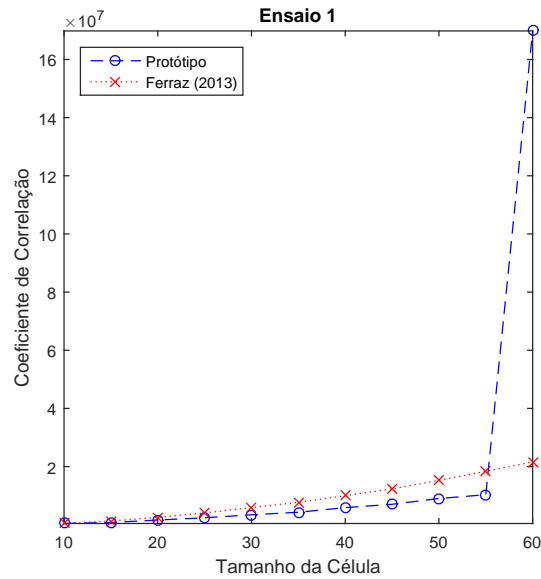
Fonte: Autor.

Figura 33 – Tempo de processamento por tamanho de célula - Exemplo 1.



Fonte: Autor.

Figura 34 – Coeficiente de correlação por tamanho de célula - Exemplo 1.



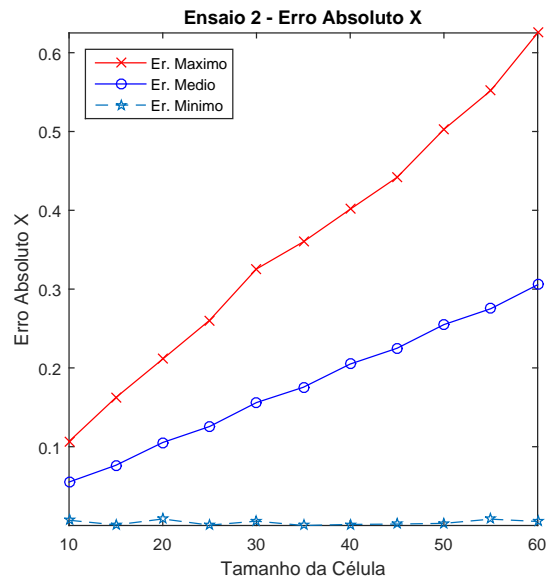
Fonte: Autor.

Os gráficos mostram que o protótipo consegue minimizar o coeficiente de correlação com mais eficiência que o algoritmo de Ferraz (2013), além de convergir em menos iterações. O protótipo apresentou, na maioria dos casos, tempo de processamento menor do que no algoritmo de Ferraz (2013). Os cálculos mais elaborados são compensados pela convergência mais rápida.

5.2.2 Exemplo 2

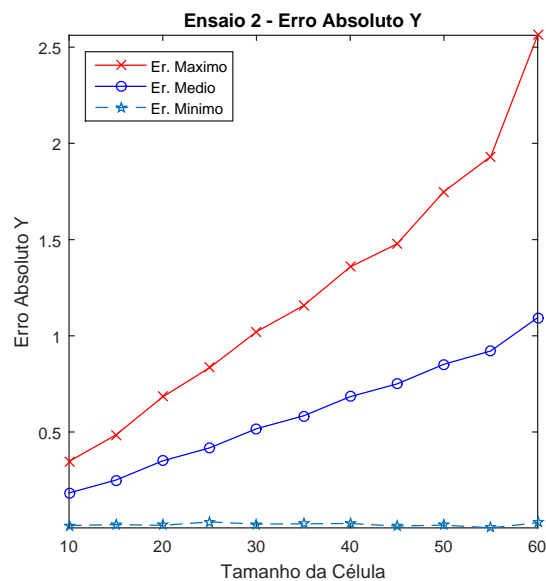
O segundo exemplo representa uma tração uniaxial em uma barra com um metro de comprimento, aplicada nas arestas verticais da imagem, conforme o modelo apresentado na Figura 35. A Tabela 3 mostra os dados utilizados para o cálculo dos deslocamentos.

Figura 36 – Erro absoluto na direção X – Exemplo 2.



Fonte: Autor.

Figura 37 – Erro absoluto na direção Y – Exemplo 2.

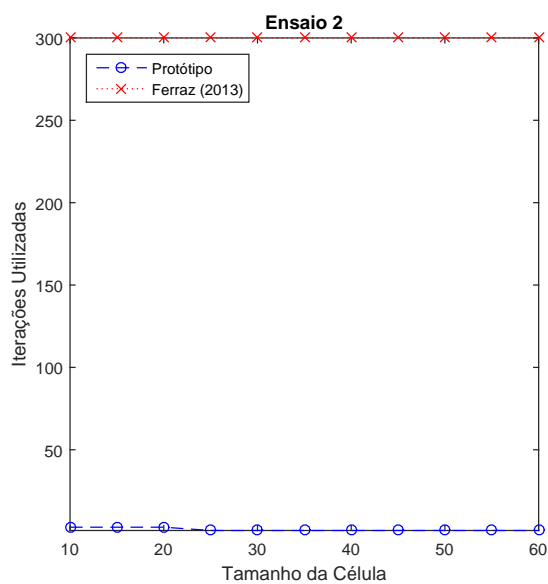


Fonte: Autor.

Para o segundo exemplo, apesar de baixos, os erros absolutos são maiores do que os erros apresentados no caso com translação de corpo rígido. Para células de tamanho máximo 60x60, os erros máximos ficam abaixo de 3,5 *pixels* para o eixo Y. No eixo X, os erros ficam abaixo de 1,5 *pixel*. O exemplo 2 apresenta campo de deslocamento de ordem 1.

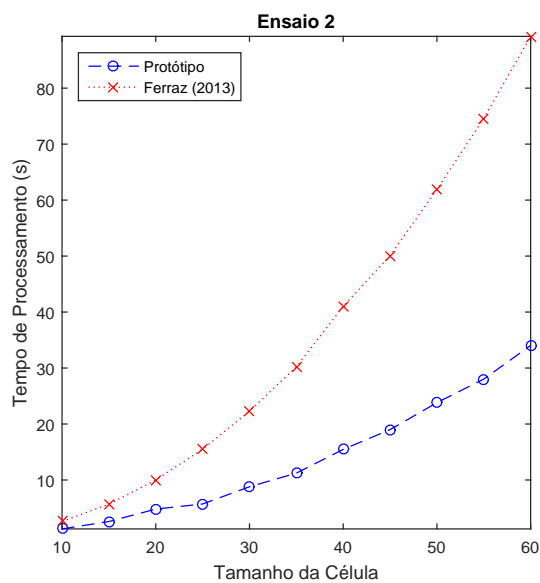
A comparação entre o protótipo e o algoritmo de Ferraz (2013) apresentou os resultados ilustrados nas Figuras 38, 39 e 40.

Figura 38 – Número de iterações por tamanho de célula – Exemplo 2.



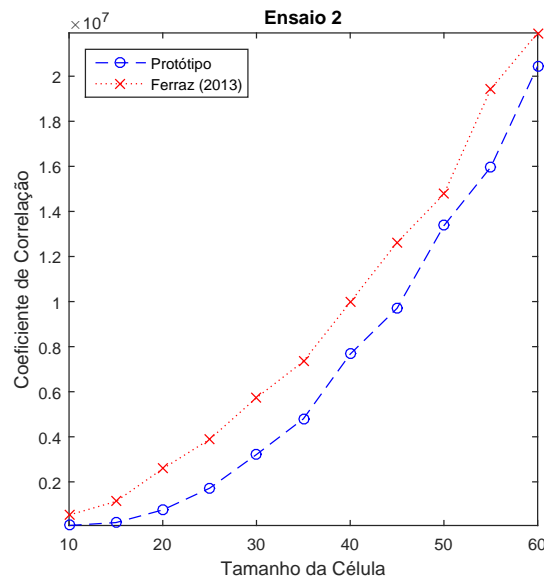
Fonte: Autor.

Figura 39 – Tempo de processamento por tamanho de célula - Exemplo 2.



Fonte: Autor.

Figura 40 – Coeficiente de correlação por tamanho de célula - Exemplo 2.



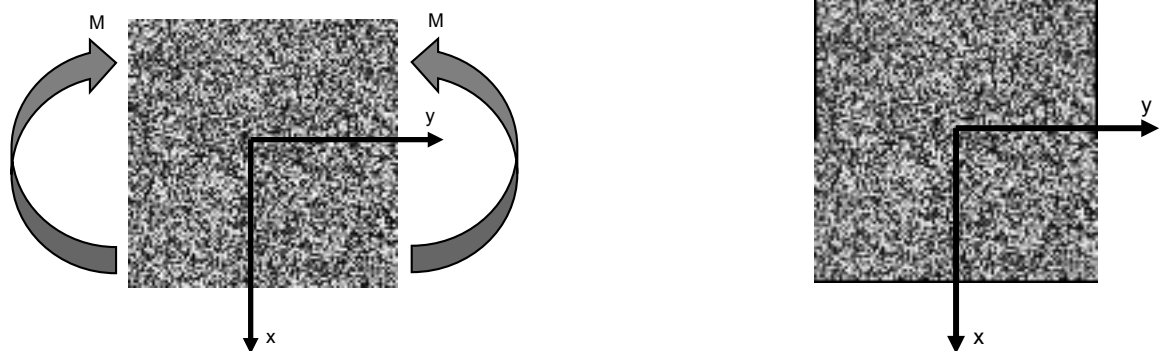
Fonte: Autor.

Novamente, percebe-se que o protótipo consegue minimizar o coeficiente de correlação com maior eficiência e, nesse caso, apresentando tempo de processamento inferior ao tempo apresentado pelo algoritmo de Ferraz (2013). A convergência do método ocorre em menos iterações: Onde o método novo utiliza menos de 15 iterações em cada situação, o algoritmo antigo atinge o critério de iterações máxima em todas as situações.

5.2.3 Exemplo 3

O terceiro exemplo consiste na aplicação de um modelo que reflete o comportamento de uma flexão reta na imagem original. O momento fletor é aplicado nas arestas verticais da imagem, como mostra a Figura 41. Os dados usados no modelo são indicados na Tabela 5.

Figura 41 – Imagens original e deformada para o exemplo 3.



(a) Modelo da Imagem Original

(b) Imagem Deformada

Fonte: Autor

Tabela 5 – Dados para o exemplo 3.

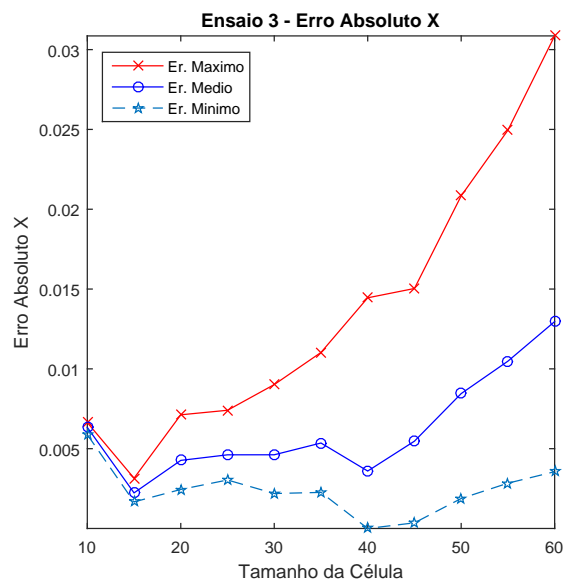
Descrição	Símbolo	Valor
Coefficiente de Poisson	μ	0,3
Momento Fletor	M	700 kNm
Módulo de Elasticidade	E	150.000 kN/m ²
Def. Normal - Dir. X	$E_{\{xx\}}$	$-\frac{M(y^2 + \nu x^2)}{2EI}$
Def. Normal - Dir. Y	$E_{\{yy\}}$	$\frac{Mxy}{EI}$
Momento de Inércia	I	1 m ⁴

A Tabela 6 mostra os parâmetros do campo de deslocamentos encontrados para as células de maior e menor tamanho, bem como para a célula de melhor correlação. Os erros absolutos apresentados pelo protótipo são apresentados nas Figuras 42 e 43.

Tabela 6 – Resultados dos parâmetros para o Exemplo 3.

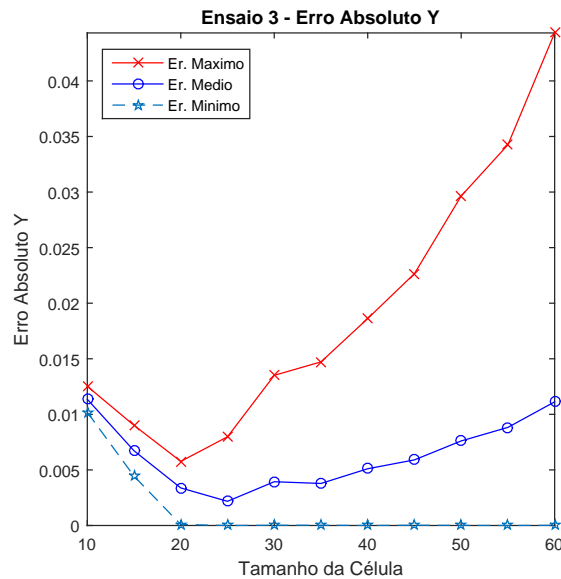
Exemplo 3	Erro Médio (px)	Parâmetros													
Melhor	0,0036	0,0029	0,0028	0	0	0	0	0	0	0	0	0	0	0	0
Maior Célula	0,0130	-0,0036	-0,0023	0	0	0	0	0	0	0	0	0	0	0	0
Menor Célula	0,0063	-0,0066	0,0144	0	0	0	0	0	0	0	0	0	0	0	0

Figura 42 – Erro absoluto na direção X – Exemplo 3.



Fonte: Autor.

Figura 43 – Erro absoluto na direção Y – Exemplo 3.



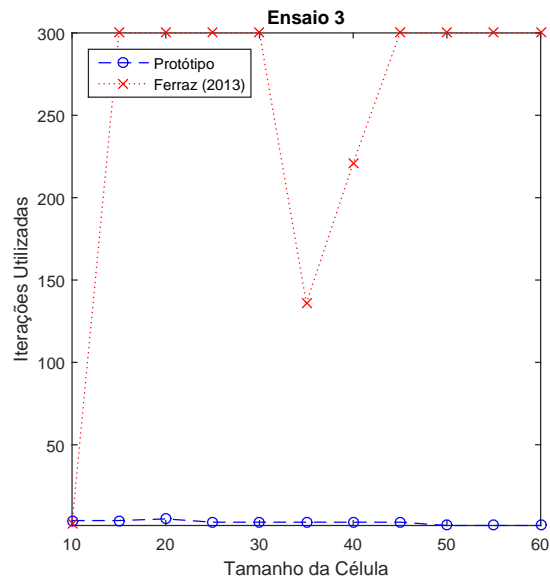
Fonte: Autor.

O protótipo apresentou excelente convergência para esse exemplo. Os maiores erros absolutos acontecem com a célula 60x60 e são inferiores a 0,05 *pixel* em ambas as direções. É interessante notar que o aumento da célula inicial gerou uma melhoria na precisão do método, significando que a célula 10x10 não apresenta informações suficientes para garantir um bom refinamento do campo.

A partir da célula 20x20, os erros aumentam pela quantidade de informações semelhantes que se encontram ao aumentar o tamanho da célula. Dessa forma, o erro máximo encontrado começa a se distanciar do erro médio. O distanciamento do erro médio indicar que a frequência do erro máximo é pequena.

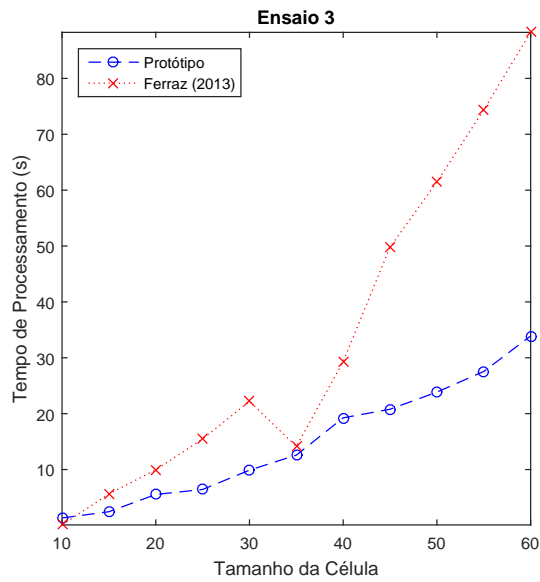
A comparação entre os dois algoritmos é ilustrada nas Figuras 44, 45 e 46.

Figura 44 – Número de iterações por tamanho de célula – Exemplo 3.



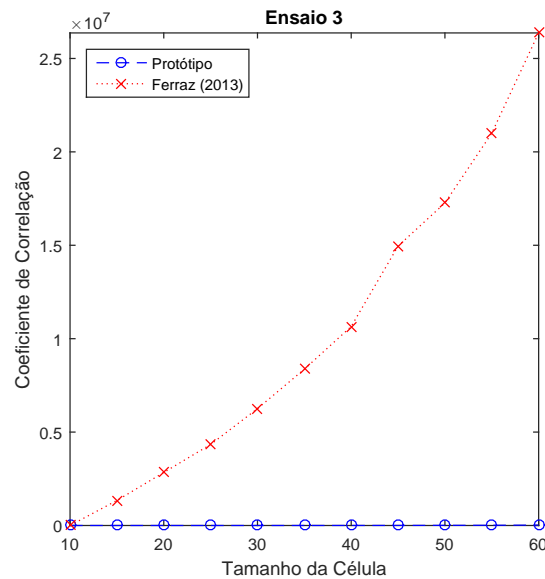
Fonte: Autor.

Figura 45 – Tempo de processamento por tamanho de célula - Exemplo 3.



Fonte: Autor.

Figura 46 – Coeficiente de correlação por tamanho de célula - Exemplo 3.



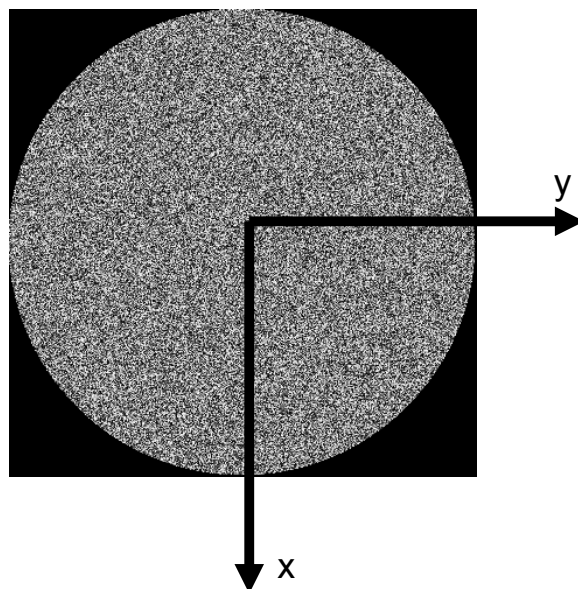
Fonte: Autor.

Novamente, assim como nos outros exemplos, o método apresentou coeficientes de correlação bem menores que aqueles calculados pelo algoritmo de Ferraz (2013). Porém, o tempo de processamento foi superior ao do algoritmo de Ferraz (2013) na maioria dos casos. Em relação à convergência dos métodos, percebe-se a rápida convergência do método atual enquanto o método anterior, em diversos momentos, atingiu o critério limitante do número de iterações máximo.

5.2.4 Exemplo 4

O quarto exemplo apresentado consiste na aplicação de um modelo de transformação de um ensaio triaxial em uma imagem. Para ilustrar melhor o "ensaio", cria-se uma imagem de um círculo, que indica a seção transversal do cilindro estudado, com um fundo em preto. A Figura 47 mostra a imagem original, que é submetida aos carregamentos de ensaio triaxial (o exemplo 4, conforme citado) e de carregamento diametral (o exemplo 5, a ser explicado em seguida). A imagem original tem 401×401 pixels. Adota-se que o círculo apresenta diâmetro de 20 cm. Assim, a escala global adotada tem 2 mm/px.

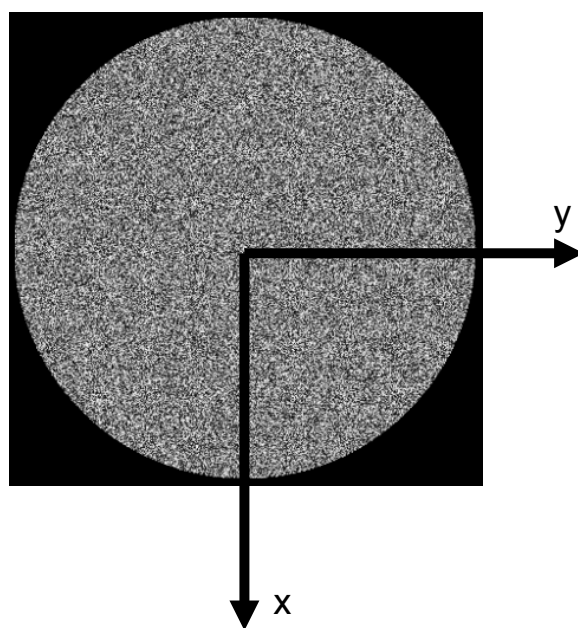
Figura 47 – Imagem original para os exemplos 4 e 5.



Fonte: Autor

O modelo para o ensaio triaxial aplicado à imagem é o mesmo apresentado por Ferraz (2013) e foi desenvolvido por Ma e Hung (2008). A imagem deformada é apresentada na Figura 48 e os dados utilizados estão na Tabela 7.

Figura 48 – Imagem deformada para o exemplo 4.



Fonte: Autor

Tabela 7 – Dados para o exemplo 4.

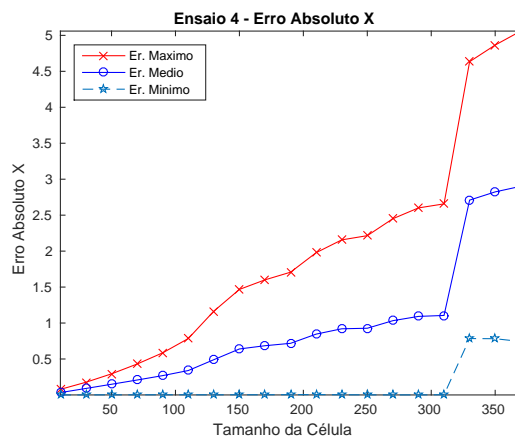
Descrição	Símbolo	Valor
Coefficiente de Poisson	ν	0,3
Módulo de Elasticidade	E	150.000 kN/m^2
Carga Distribuída	P	5.000 kN/m^2
Deformação Normal na Direção X	E_{xx}	$-\frac{(1-\nu)P\sqrt{x^2+y^2}}{yE\sqrt{\frac{x^2+y^2}{y^2}}} = -\frac{(1-\nu)P}{E}$
Deformação Normal na Direção Y	E_{yy}	$-\frac{(1-\nu)P\sqrt{x^2+y^2}}{xE\sqrt{\frac{x^2+y^2}{x^2}}} = -\frac{(1-\nu)P}{E}$

Para o exemplo de aplicação do ensaio de compressão triaxial, o método apresentou os resultados ilustrados nas Figuras 49 e 50. A Tabela 8 mostra os parâmetros do campo de deslocamentos encontrados para as células de maior e menor tamanho. Nesse caso, a célula de menor tamanho também apresentou melhor correlação.

Tabela 8 – Resultados dos parâmetros para o Exemplo 4.

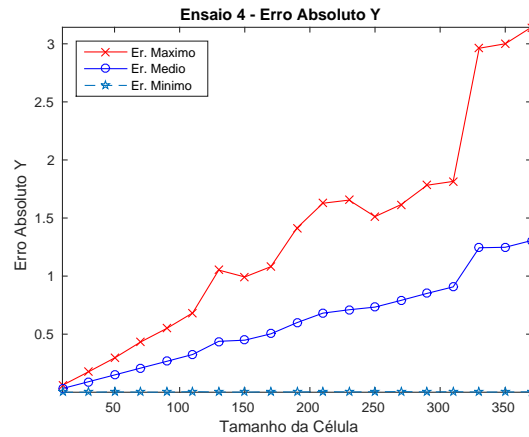
Exemplo 4	Erro Médio (px)	Parâmetros											
Melhor	0,0331	-0,0110	0	0	0	0	0	0	0	0	0	0	0
Maior Célula	2,9014	2,9014	-0,9837	0	0	0	0	0	0	0	0	0	0

Figura 49 – Erro absoluto na direção X – Exemplo 4.



Fonte: Autor.

Figura 50 – Erro absoluto na direção Y – Exemplo 4.



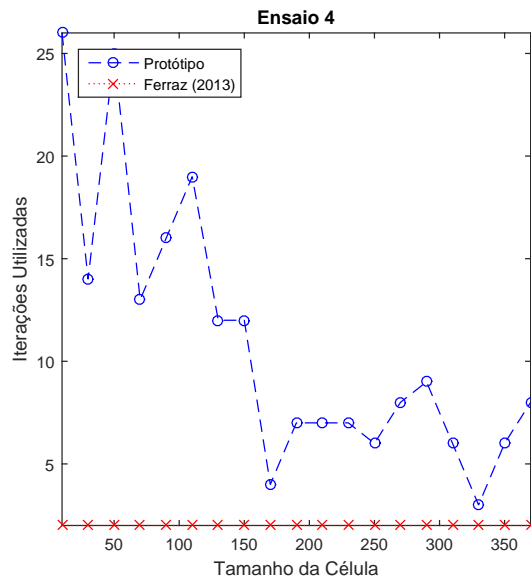
Fonte: Autor.

Percebe-se que o método apresentou valores de erros máximos muito acima dos valores dos erros médios. Essas ocorrências são relativamente raras e acontecem em *pixels* mais distantes das células. Retirando-se esse valor, o erro absoluto máximo apresentado nas direções X e Y é um pouco menor que 5 *pixels*. Para os casos dos exemplos 4 e 5, a escala é de 2 mm por *pixels*. Assim, o erro máximo, em mm, é de 10 mm na direção X, em um cilindro com 20 cm de diâmetro.

Na comparação entre algoritmos, o protótipo convergiu mais tardiamente e com um tempo de processamento maior do que o ambiente inicial. No entanto, a minimização dos coeficientes de correlação com o aumento da célula se manteve melhor.

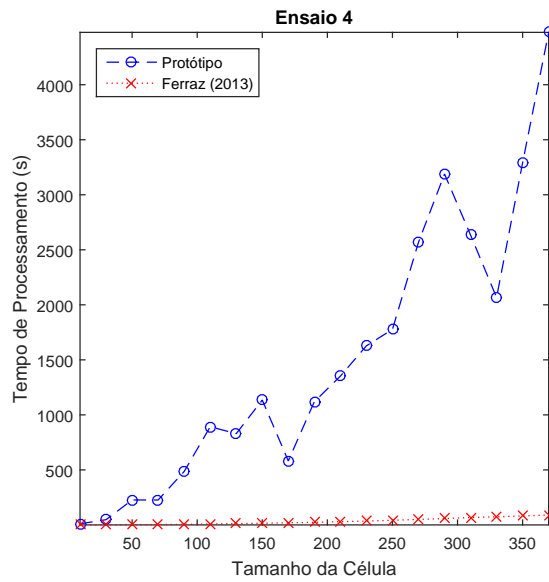
Entende-se que, para o caso de uma compressão triaxial, onde há tendência do acúmulo de *pixels* em posições muito próximas, o ambiente computacional de alto desempenho precisa realizar mais iterações para encontrar os valores adequados. As Figuras 51, 52 e 53 mostram os valores obtidos.

Figura 51 – Número de iterações por tamanho de célula – Exemplo 4.



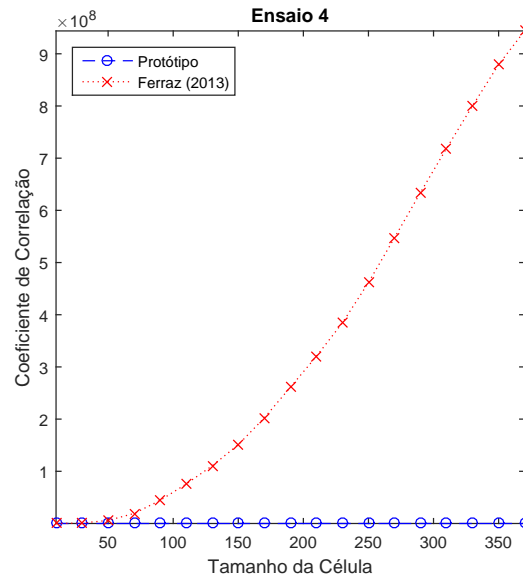
Fonte: Autor.

Figura 52 – Tempo de processamento por tamanho de célula - Exemplo 4.



Fonte: Autor.

Figura 53 – Coeficiente de correlação por tamanho de célula - Exemplo 4.



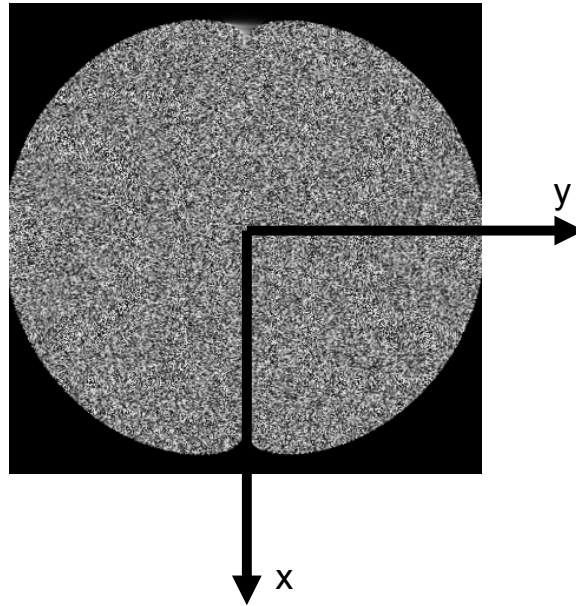
Fonte: Autor.

5.2.5 Exemplo 5

O último exemplo consiste em um corpo cilíndrico sendo submetido a um carregamento aplicado diametralmente. O ensaio agora estudado, denominado Ensaio Lobo Carneiro, foi criado para determinação indireta da resistência do concreto à tração. O modelo matemático, também utilizado por Ferraz (2013), também é apresentado por Ma e Hung (2008) também.

A Figura 54 mostra a imagem deformada e a Tabela 9 contém os dados utilizados no modelo. Como o modelo analítico apresenta singularidade no ponto de aplicação da carga, o deslocamento dos *pixels* nessa região gera um formato específico.

Figura 54 – Imagem deformada para o exemplo 5. Fonte: Autor.



Fonte: Autor.

Tabela 9 – Dados para o exemplo 5.

Descrição	Símbolo	Valor
Coeficiente de Poisson	ν	0,3
Módulo de Elasticidade	E	150.000 kN/m^2
Carga Distribuída	P	5.000 kN/m^2

As expressões para o cálculo dos deslocamentos u_r e u_θ em coordenadas polares foram retiradas de Ma e Hung (2008) e seguem apresentadas nas Equações 5.1 e 5.2.

$$u_r = -\frac{2P}{E\pi t} \left\{ \left[-\frac{2 + \rho^2 + \rho^4 - (1 + 3\rho^2) \cos(2\theta)}{1 + \rho^4 - 2\rho^2 \cos(2\theta)} + \sin(\theta)(\tan^{-1}(\gamma) + \tan^{-1}(\beta)) \right] + \cos(\theta) \ln \left| \frac{1 + \rho^2 + 2\rho \cos(\theta)}{1 + \rho^2 - 2\rho \cos(\theta)} \right| \right\} + \nu \left[-\frac{\rho(1 - \rho^2)(\rho^2 - \cos(2\theta))}{1 + \rho^4 - 2\rho^2 \cos(2\theta)} - \sin(\theta)(\tan^{-1}(\gamma) + \tan^{-1}(\beta)) \right] \quad (5.1)$$

$$u_\theta = -\frac{2P}{E\pi t} \left\{ \left[\cos(\theta)(\tan^{-1}(\gamma) + \tan^{-1}(\beta)) - \sin(\theta) \left(\frac{2\rho(1 - \rho^2) \cos(\theta)}{1 + \rho^4 - 2\rho^2 \cos(2\theta)} \right) \right] + \ln \left| \frac{1 + \rho^2 + 2\rho \cos(\theta)}{1 + \rho^2 - 2\rho \cos(\theta)} \right| \right\} - \nu \cos(\theta) \left[\frac{2\rho(1 - \rho^2) \cos(\theta)}{1 + \rho^4 - 2\rho^2 \cos(2\theta)} + \tan^{-1}(\gamma) + \tan^{-1}(\beta) \right] \quad (5.2)$$

As respectivas variáveis apresentadas na equação acima são definidas

$$\rho = \frac{1}{R} \sqrt{x^2 + y^2} \tag{5.3a}$$

$$\theta = \tan^{-1} \left(\frac{x}{y} \right) \tag{5.3b}$$

$$\beta = \frac{\rho \sin(\theta)}{1 - \rho \cos(\theta)} \tag{5.3c}$$

$$\gamma = \frac{\rho \sin(\theta)}{1 + \rho \cos(\theta)} \tag{5.3d}$$

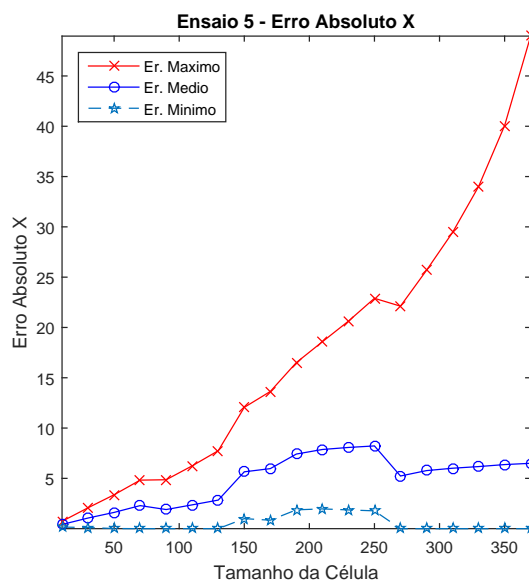
O exemplo 5 apresenta o campo de deslocamentos mais complexo em relação aos outros campos aplicados nas outras imagens. O ensaio de compressão diametral apresentou os erros absolutos, demonstrados nas Figuras 55 e 56.

A Tabela 10 mostra os parâmetros do campo de deslocamentos encontrados para as células de maior e menor tamanho. Nesse caso, a célula de menor tamanho também apresentou melhor correlação.

Tabela 10 – Resultados dos parâmetros para o Exemplo 5.

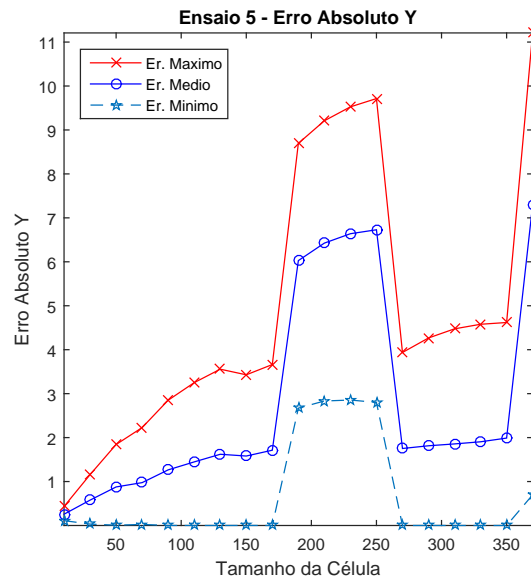
Exemplo 5	Erro Médio (px)	Parâmetros											
Melhor	0,4164	-0,0180	-0,0296	0	0	0	0	0	0	0	0	0	0
Maior Célula	6,4960	0,5472	-3,584	0	0	0	0	0	0	0	0	0	0

Figura 55 – Erro absoluto na direção X – Exemplo 5.



Fonte: Autor.

Figura 56 – Erro absoluto na direção Y – Exemplo 5.



Fonte: Autor.

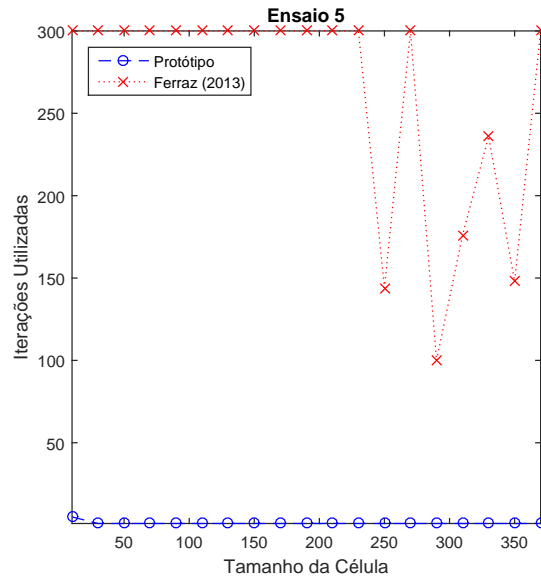
Os erros aumentam com o tamanho da célula, mantendo-se a um nível absoluto máximo de 5 *pixels* (ou 1 cm) em regiões de tamanho máximo 100x100. O crescimento dos erros com o aumento da célula mostra que a quantidade de informações analisadas pelo método acaba por gerar dificuldades na convergência, como visto no gráfico com a quantidade de iterações utilizada para o refino.

Percebe-se que há um distanciamento dos erros absolutos máximos em relação aos erros médios e mínimos com o tamanho da célula. Entende-se que o aumento do raio em relação ao centro do círculo apresenta valores cada vez mais discrepantes da tendência normal.

Para a comparação com os resultados de Ferraz (2013), as Figuras 57, 58 e 59 mostram que o tempo de processamento do algoritmo atual apresentou valores menores entre os tempos gastos pelo algoritmo inicial com o aumento da célula de análise. Além disso, a convergência do algoritmo de Ferraz (2013) foi mais rápida.

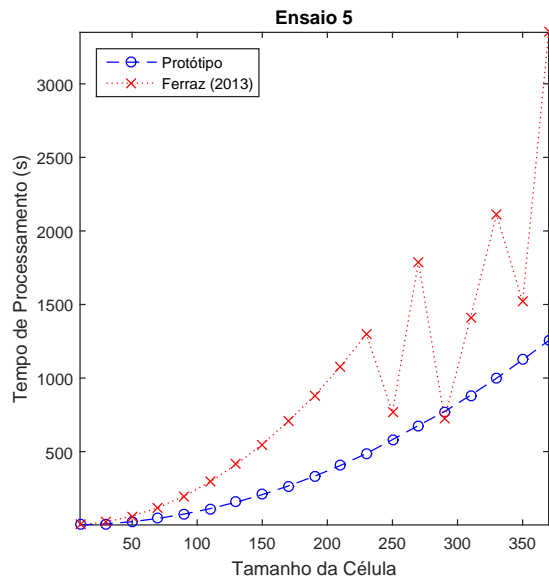
A correlação, por outro lado, não foi tão eficiente quanto nos casos anteriores, mantendo-se próximo ao coeficiente calculado pelo algoritmo de Ferraz (2013). Atribui-se a situação às condições de contorno nos quais o modelo deformado foi gerado. A presença da carga pontual gerou deslocamentos relativos muito grandes nos pontos de aplicação. Por outro lado, o algoritmo teve dificuldades em encontrar um mínimo para um campo de deslocamentos muito irregular.

Figura 57 – Número de iterações por tamanho de célula – Exemplo 5.



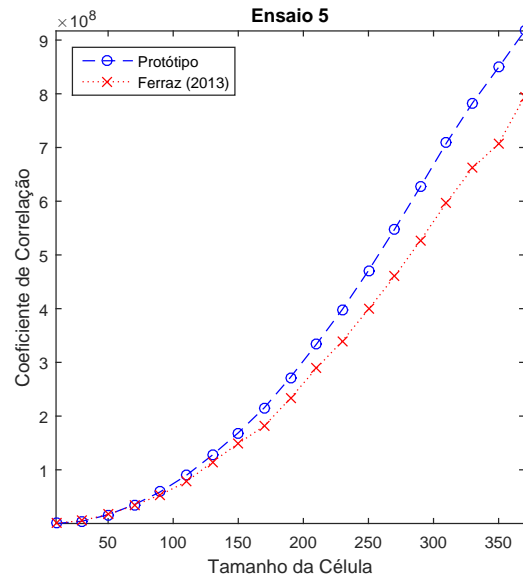
Fonte: Autor.

Figura 58 – Tempo de processamento por tamanho de célula - Exemplo 5.



Fonte: Autor.

Figura 59 – Coeficiente de correlação por tamanho de célula - Exemplo 5.

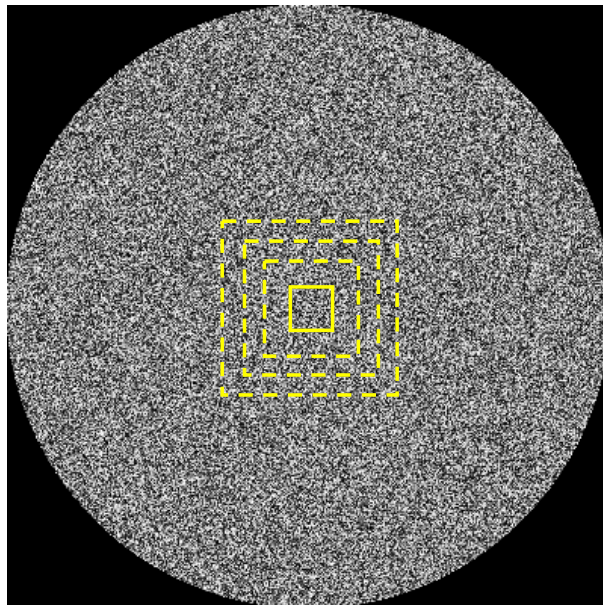


Fonte: Autor.

5.3 Computação de Alto Desempenho

Na seção anterior, os ensaios determinavam o deslocamento em um único ponto, localizado no centro da imagem, enquanto o tamanho da célula usada para o cálculo da correlação variava. Para cada exemplo, foram calculados os erros absolutos, bem como a eficiência do protótipo em relação ao algoritmo de Ferraz (2013). A Figura 60 representa um exemplo da região de análise.

Figura 60 – Escopo da análise realizada na Seção 5.



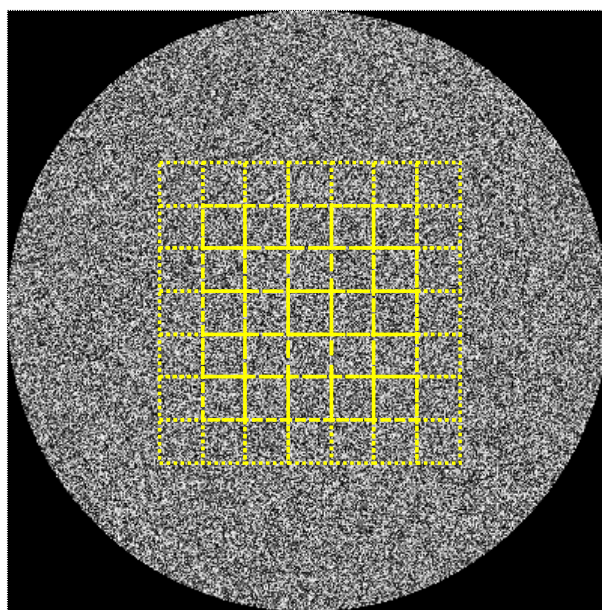
Fonte: Autor.

O deslocamento de um único ponto não permite a determinação do campo de deslocamentos de uma região. Uma estratégia que pode ser usada consiste na determinação pontual dos deslocamentos em diversos pontos distribuídos regularmente no domínio da imagem. A combinação dos resultados desses pontos através de uma técnica de interpolação pode determinar de forma aproximada o campo de deslocamentos da região estudada.

Como se pode notar nos gráficos comparativos dos desempenhos apresentados na seção anterior, há casos onde o custo computacional de análise foi de aproximadamente 4000 segundos (1 hora, aproximadamente) para analisar uma única célula. Portanto, existe a necessidade de aplicação de técnicas de Computação de Alto Desempenho (*High Processing Computing* - HPC) para acelerar o processamento de análise para uma quantidade grande de pontos. A presente seção compara o desempenho computacional ao usar HPC.

As análises em questão são aplicadas aos mesmos exemplos da seção anterior. Nesta Seção, avaliam-se apenas informações relativas ao desempenho computacional nessa seção, uma vez que a precisão do protótipo foi apresentada anteriormente. Geram-se malhas quadradas de células com tamanho pré-definido. A quantidade de células por grade varia da mesma forma nas duas direções. A Figura 61 exemplifica as grades geradas para a análise.

Figura 61 – Grade de células para avaliação de desempenho.



Fonte: Autor.

Para cada ensaio, é medido o tempo computacional de análise com e sem a aplicação de técnicas de HPC. Com o aumento da ordem da malha do problema (crescimento da quantidade de células), consegue-se calcular os índices de Escalabilidade e *Speed-up* para cada exemplo.

O conceito de escalabilidade, conforme apresentado por Bondi (2000), é a habilidade de um sistema em acomodar um aumento no número de elementos, processando maiores quanti-

dades de trabalho sem gerar problemas. Dessa forma, a escalabilidade é calculada relacionando, para cada ordem do *grid*, o seu respectivo *speed-up*.

Speed-up (SUP) é definido por Xian-He S. e Rover (1994) como a relação entre o tempo gasto para executar uma operação usando HPC (T_{HPC}) e o tempo gasto para executar uma operação sem usar HPC (T_{SEQ})

$$SUP = \frac{T_{HPC}}{T_{SEQ}}. \quad (5.4)$$

Para os estudos de casos 1 a 3, onde o tamanho da imagem é de 101 x 101 *pixels*, o tamanho definido para a célula é de 20 x 20 *pixels*. As malhas variam de 2x2 até 5x5 células. Para os estudos de casos 4 e 5, cujo tamanho da imagem é de 401 x 401 *pixels*, o tamanho da célula é 55 *pixels*. As malhas analisadas variam de 2x2 até 6x6 células.

Para cálculo da curva do *speed-up*, a malha usada é de 4x4 nos estudos de casos 1 a 3 e 6x6 nos estudos de casos 4 e 5. As análises realizadas com HPC usaram de 2 a 12 processadores.

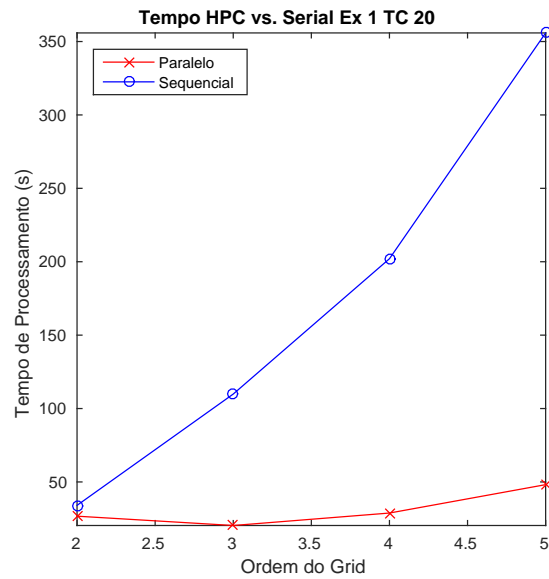
A técnica de HPC usada está inclusa no *toolkit* de processamento paralelo do Matlab (The MathWorks Inc., 2015). Esse *toolkit* usa técnicas de memória compartilhada (onde um computador possui mais de um processador) para realizar os cálculos.

A máquina usada para a análise em paralelo possui 32 núcleos (16 físicos e 16 virtualizados), 128 GBs de memória RAM com sistema operacional Linux.

5.3.1 Exemplo 1

Sendo a mais simples das transformações, a análise da translação de corpo rígido é realizada rapidamente, quando comparada com os exemplos seguintes. Mesmo ao tratar uma malha com 25 células, o algoritmo leva aproximadamente 350 segundos para finalizar a análise. O uso de técnicas de HPC diminuiu o tempo de análise para 50 segundos, aproximadamente, como mostra a Figura 62.

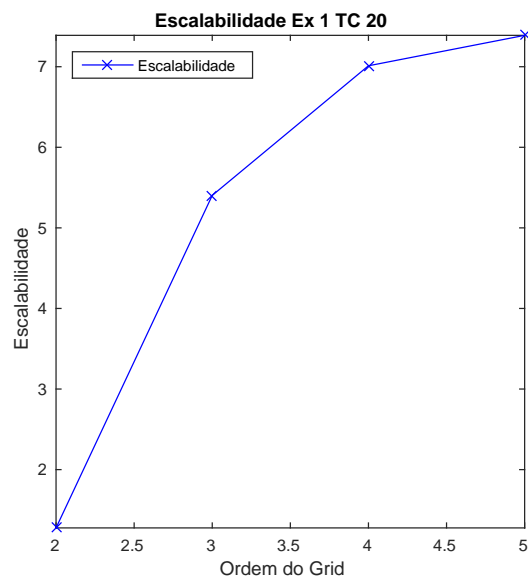
Figura 62 – Tempo computacional para o exemplo 1.



Fonte: Autor.

O uso de técnicas de HPC para as análises mostra ganhos de tempo computacional superiores a 7 vezes o tempo sequencial. O uso de HPC com malhas menores resultou em ganhos significativos que foram diminuindo com o aumento da carga de trabalho. Essa situação indica, logicamente, que o aumento da quantidade de análises implica no aumento do tempo computacional, mesmo com a divisão de análises em mais processadores. A Figura 63 apresenta a escalabilidade do exemplo.

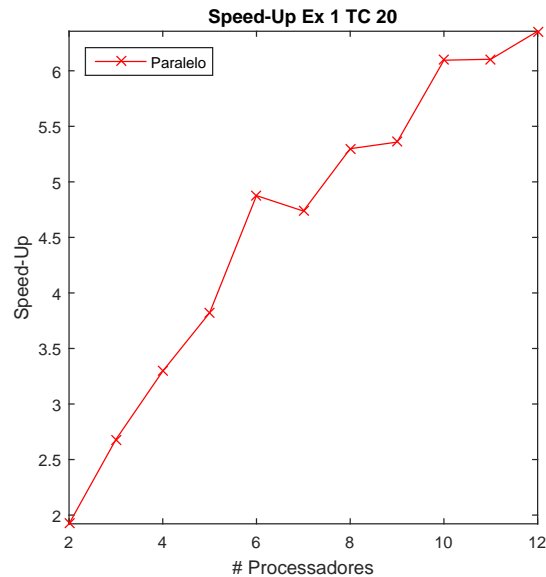
Figura 63 – Escalabilidade para o exemplo 1.



Fonte: Autor.

O cálculo do *speed-up* obtém ganhos superiores a 8x, conforme apresentado na Figura 64. O uso de mais processadores aumenta a divisão da carga de trabalho, diminuindo o tempo computacional gasto para a execução da análise.

Figura 64 – *Speed-up* para o exemplo 1.

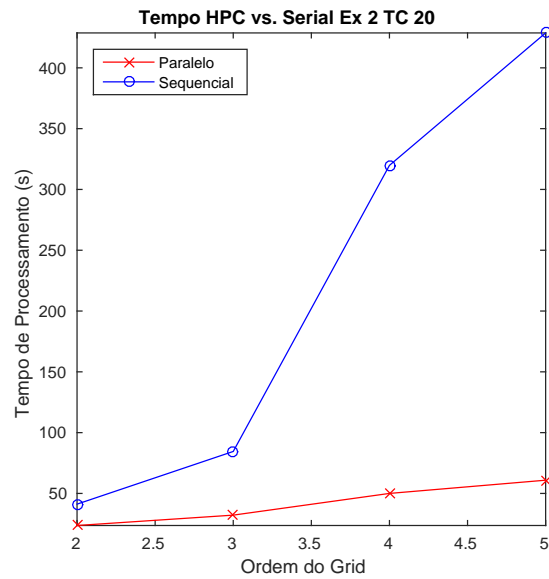


Fonte: Autor.

5.3.2 Exemplo 2

A análise do campo para tração axial apresentou tempo computacional mais elevado que a translação de corpo rígido. Esse tipo de transformação usa termos lineares de deslocamentos, o que justifica o aumento no tempo computacional. O tempo máximo que o algoritmo leva é 430 segundos, aproximadamente. A utilização de HPC diminuiu o tempo de análise para 55 segundos, conforme a Figura 65.

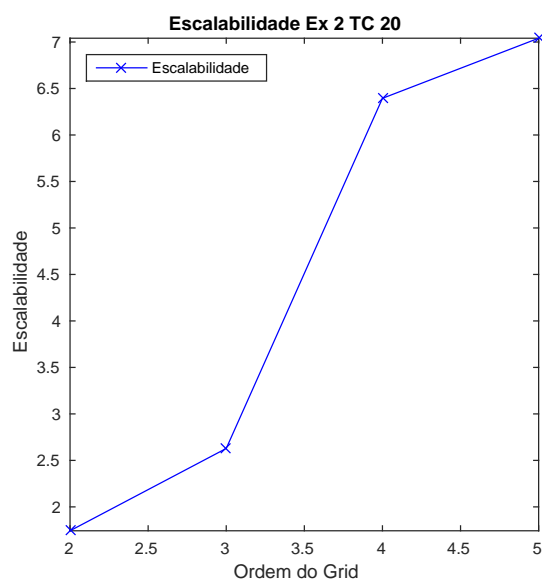
Figura 65 – Tempo computacional para o exemplo 2.



Fonte: Autor.

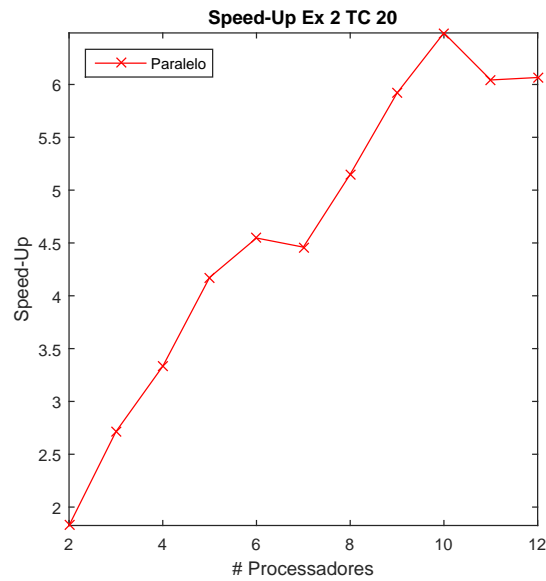
A escalabilidade, neste exemplo, apresentou maior ganho quando a ordem da malha saiu de 3x3 para 4x4, chegando ao valor máximo de 7x de ganho de performance com o aumento de carga de análise. A Figura 66 apresenta os resultados da escalabilidade.

Figura 66 – Escalabilidade para o exemplo 2.



Fonte: Autor.

O *speed-up* apresentou valor reduzido em comparação com o exemplo anterior, alcançando valores máximos próximos a 6,5x de ganho em tempo. A Figura 67 mostra os resultados para o *speed-up*.

Figura 67 – *Speed-up* para o exemplo 2.

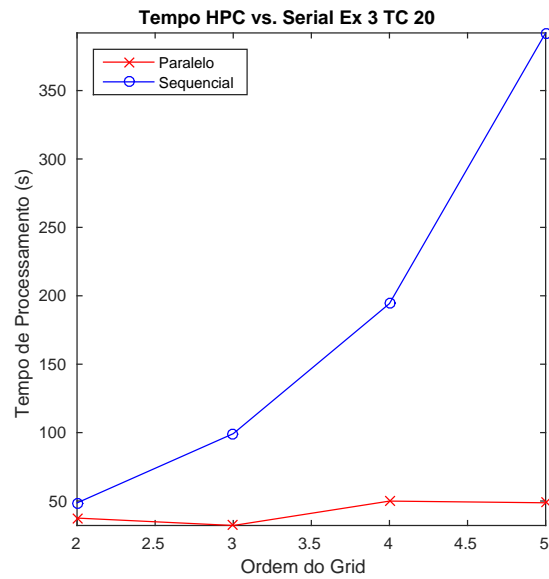
Fonte: Autor.

5.3.3 Exemplo 3

A função de transformação para flexão pura possui termos quadráticos em sua composição. Apesar disso, o custo computacional máximo de análise é inferior ao custo máximo para o exemplo anterior. O tempo máximo de análise desse exemplo foi de 380 segundos, aproximadamente. A Figura 68 mostra os resultados do tempo computacional para esse exemplo.

Na seção anterior, foi visto que esse exemplo apresentou os menores valores de erros absolutos nos dois eixos. Acredita-se que esses bons resultados decorram do método de geração de imagens, pois os *pixels* não apresentaram uma variação muito grande na sua posição.

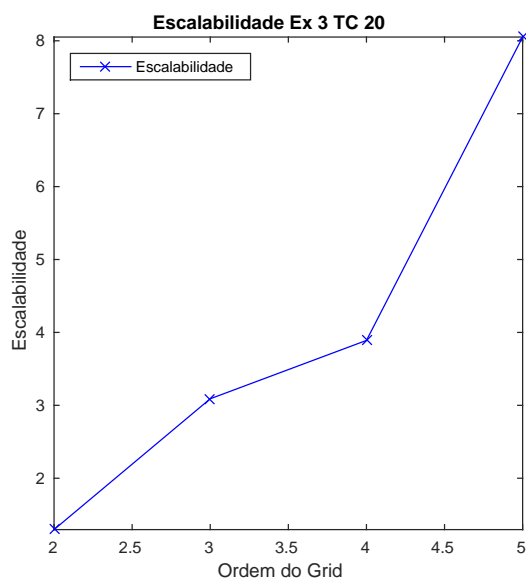
Figura 68 – Tempo computacional para o exemplo 3.



Fonte: Autor.

Em termos de escalabilidade, este exemplo apresentou ganhos máximos semelhantes àqueles apresentados no exemplo 1. Nesse caso, o ganho máximo é um pouco superior à 8x, conforme pode ser visualizado na Figura 69. A maior variação do ganho ocorre na última mudança da ordem, quando a malha passou a ter 25 células (5x5).

Figura 69 – Escalabilidade para o exemplo 3.

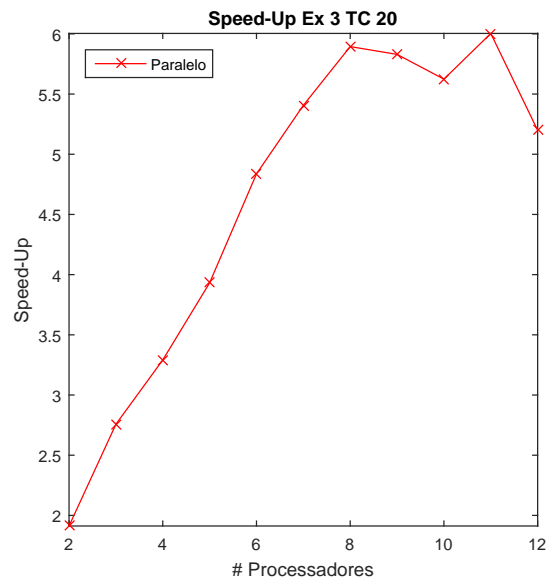


Fonte: Autor.

O *speed-up* máximo para esse exemplo foi 6x. A declividade da curva mostra que ele conseguiu atingir valores próximos do pico rapidamente. Para o caso, o uso de 8 processadores

foi suficiente para gerar ganhos computacionais próximos do máximo calculado. A Figura 70 mostra os resultados gerados.

Figura 70 – *Speed-up* para o exemplo 3.

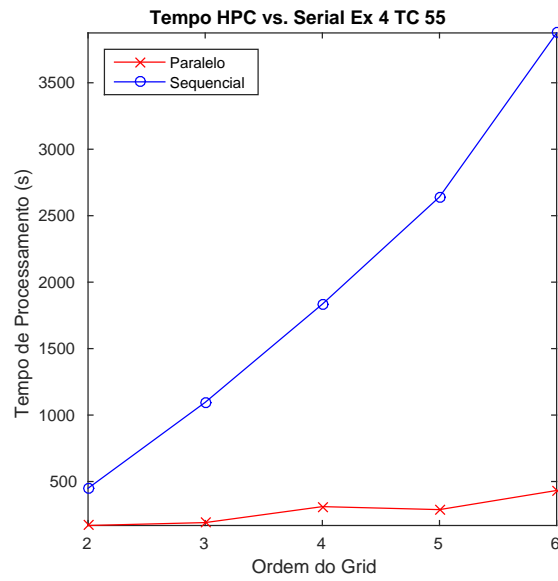


Fonte: Autor.

5.3.4 Exemplo 4

A função de transformação para esse exemplo possui uma complexidade maior do que os exemplos anteriores, o que se comprova no tempo máximo necessário para finalizar a análise usando apenas um processador. O custo computacional máximo encontrado é superior a 3700 segundos, enquanto que o uso de técnicas de HPC diminuiu esse tempo para 500 segundos, aproximadamente. A Figura 71 apresenta os valores do tempo computacional gasto na análise.

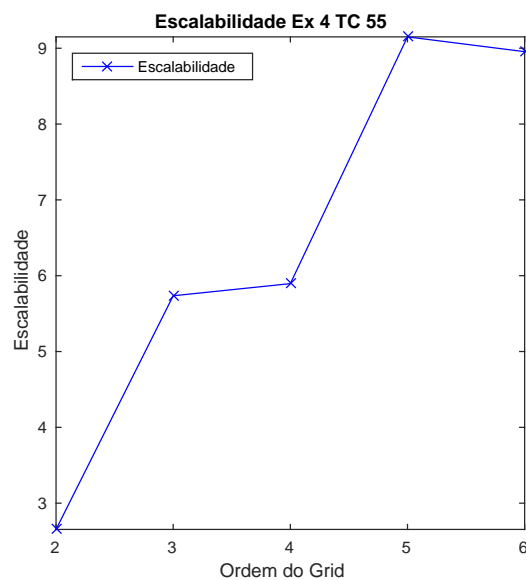
Figura 71 – Tempo computacional para o exemplo 4.



Fonte: Autor.

A escalabilidade alcançou os valores mais altos dentre os estudos de casos analisados. Para uma malha com 25 células, o valor máximo foi um pouco superior a 9x, como mostrado na Figura 72.

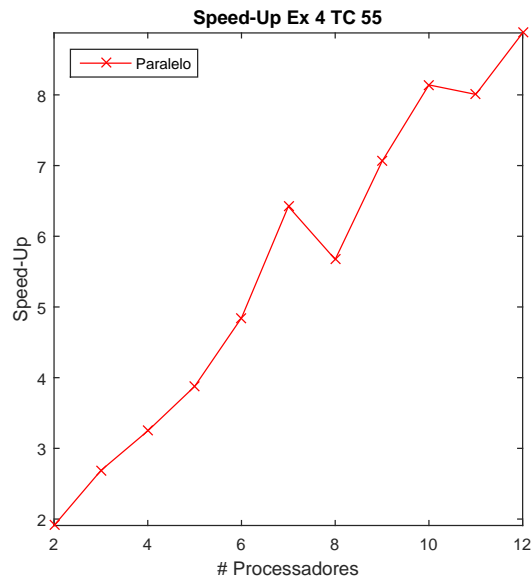
Figura 72 – Escalabilidade para o exemplo 4.



Fonte: Autor.

O *speed-up* também apresentou valores relativamente altos, com ganho máximo próximo de 9x. A Figura 73 apresenta os valores de ganhos em tempo para esse exemplo.

Figura 73 – *Speed-up* para o exemplo 4.

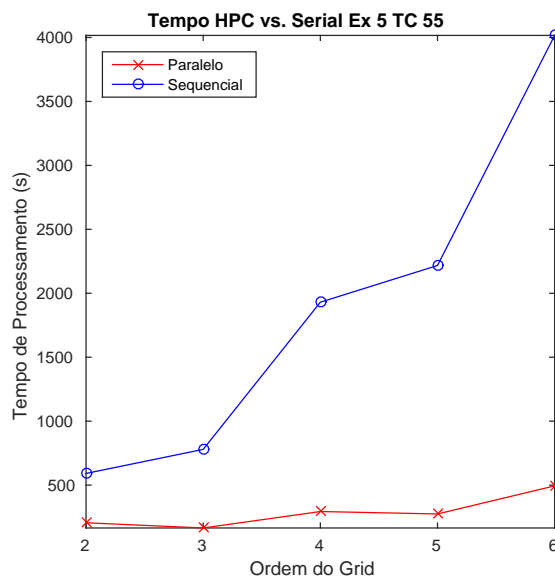


Fonte: Autor.

5.3.5 Exemplo 5

O exemplo com o campo de deslocamentos mais complexo apresentou custo computacional máximo de aproximadamente 4000 segundos para cálculo do deslocamento de 36 pontos. Ao utilizar HPC, o tempo foi reduzido para 500 segundos. A Figura 74 mostra os tempos computacionais gastos.

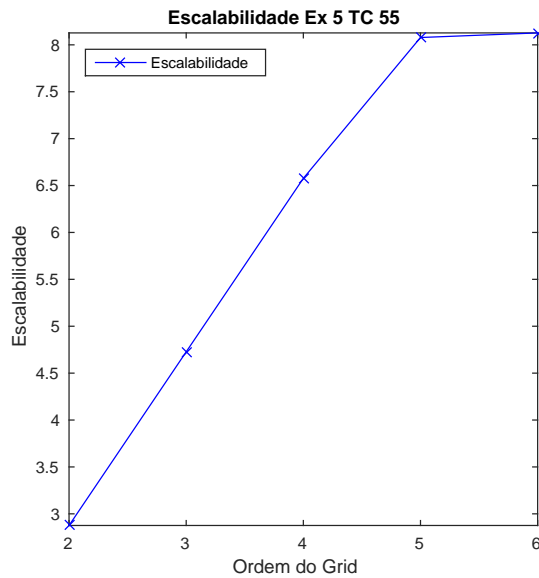
Figura 74 – Tempo computacional para o exemplo 5.



Fonte: Autor.

A escalabilidade do problema atingiu um valor máximo superior a 8x na malha 6x6 e um ganho muito próximo desse valor na malha 5x5. Pela diferença de declividade apresentado na Figura 75, percebe-se que a escalabilidade da técnica atingiu um patamar de uso onde o aumento de carga não deve gerar maiores ganhos.

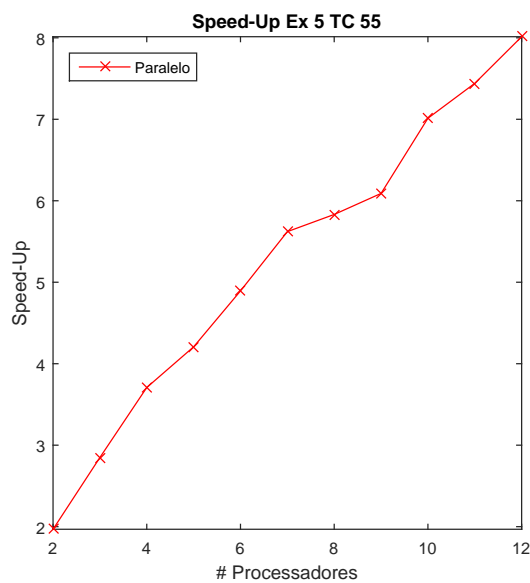
Figura 75 – Escalabilidade para o exemplo 5.



Fonte: Autor.

A curva de *speed-up* apresenta ganhos de 8x, com um crescimento relativamente consistente. A Figura 76 apresenta a curva de *speed-up* para o exemplo 5.

Figura 76 – *Speed-up* para o exemplo 5.



Fonte: Autor.

Os dados obtidos em todos os exemplos mostram a vantagem da incorporação de técnicas de computação de alto desempenho. Em todos os casos, o custo computacional diminuiu bastante, alcançando o custo máximo aproximado de 500 segundos para análise de 36 células de uma imagem para um campo de deslocamentos relativamente complexo. Assim, a aplicação de técnicas de HPC é altamente recomendada para a diminuição do custo computacional de análise usando correlação de imagens digitais a partir de uma abordagem local.

5.4 Método SIFT

5.4.1 Procedimento de Análise

O procedimento de análise fez uso do algoritmo implementado por Rey Otero e Delbracio (2014), que apresentaram uma solução *web* com a descrição e o algoritmo método SIFT. Essa solução foi usada para a obtenção dos pontos característicos. A calibração do algoritmo para a busca dos pontos de características invariantes é realizada usando os parâmetros *default* do algoritmo, conforme apresentados a seguir. Há uma página na solução onde esses parâmetros podem ser alterados.

- Detecção das Características
 - Número de *octaves*: 8
 - Número de escalas por *octaves*: 3
 - Nível mínimo de borramento no espaço-escala: 0,8
 - Superamostragem com fator: 2
 - Nível de borramento assumido na imagem de entrada: 0,5
 - Limite para operação Diferença do Gaussiano: 0,01
 - Limite na razão das curvaturas principais: 10

- Descrição
 - Número de colunas no histograma de orientação: 36
 - Desvio padrão da janela gaussiana para atribuição da orientação: 1,5
 - Limite para referências secundárias para orientação: 0,8
 - Ordem da matriz de histogramas de orientação para constituir o descritor: 4
 - Quantidade de colunas no histograma ponderado: 8
 - Desvio padrão da janela gaussiana para descritor: 6

- Correspondência com limite relativo

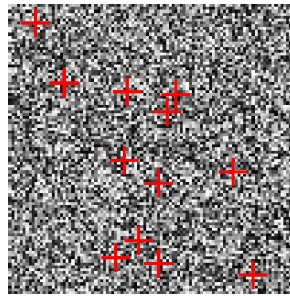
- Limite na razão da distância: 0.6

As coordenadas dos pontos encontradas pelo algoritmo de Rey Otero e Delbracio (2014) são submetidas ao protótipo DIC para comparação entre os pontos encontrados pelos dois métodos. Os deslocamentos encontrados pelos dois métodos são comparados com os deslocamentos de referência, calculados usando os modelos de transformação usados na geração de imagens. A diferença dos valores calculados pelos métodos DIC e SIFT em relação aos valores de referência são apresentados na seção a seguir.

5.4.2 Resultados

Os exemplos usados são os mesmos descritos anteriormente, nesta mesma Seção. Para o primeiro exemplo, o método SIFT detectou alguns pontos com características invariantes. A Figura 77 apresenta a localização dos pontos na imagem.

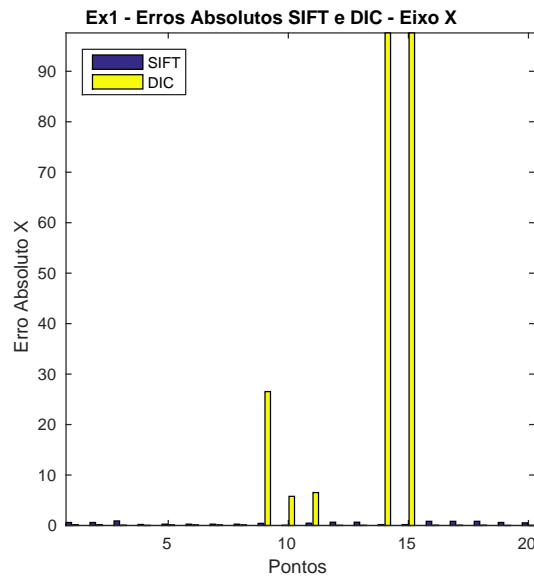
Figura 77 – Pontos detectados pelo método SIFT - Exemplo 1.



Fonte: Autor.

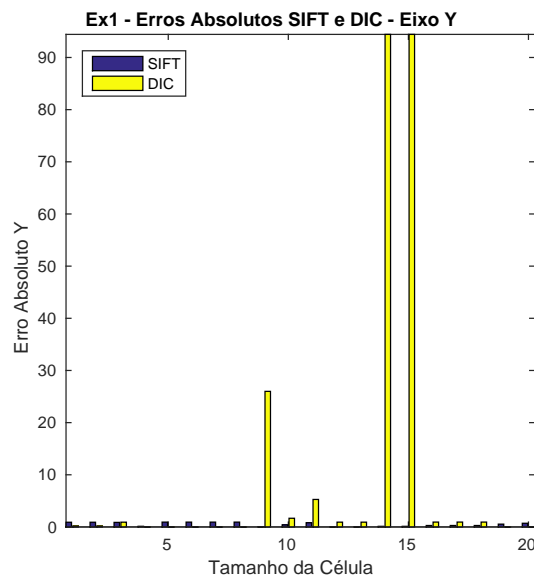
Os resultados obtidos nos métodos DIC e SIFT são comparados com os valores de referência. Para cada ponto, é calculado o erro absoluto em cada eixo coordenado. Os resultados são apresentados nas Figuras 78 e 79. Pode-se perceber que em alguns pontos os erros obtidos pelo método DIC são muito maiores do que aqueles apresentados pelo método SIFT. Isso decorre da escolha dos pontos, que é feita pelo método SIFT. Como o método DIC depende mais das informações da vizinhança, ele pode aceitar um ponto cuja correlação não seja muito alta ou aceitar um ponto de alta correlação, mas em uma localização diferente. De uma maneira geral, os dois métodos apresentaram valores próximos para a maioria dos pontos detectados no método SIFT.

Figura 78 – Erros absolutos entre SIFT e DIC para o eixo X - Exemplo 1.



Fonte: Autor.

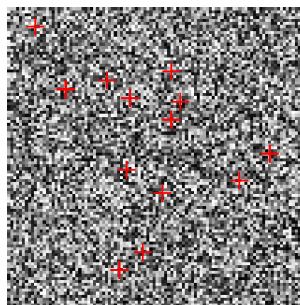
Figura 79 – Erros absolutos entre SIFT e DIC para o eixo Y - Exemplo 1.



Fonte: Autor.

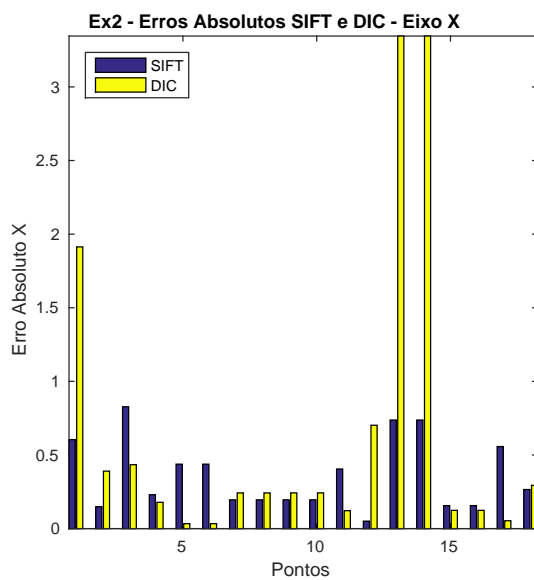
No segundo exemplo, a localização dos pontos obtidos por SIFT está descrita na Figura 80. Pode-se ver que, assim como no primeiro exemplo, alguns pontos apresentaram discrepâncias muito altas, com várias ocasiões onde o método SIFT apresentou menos erros que o método DIC. As discrepâncias ocorrem, pois o método DIC encontra um ponto com correlação semelhante, mas que não é o ponto correto. Percebe-se também pontos onde o método DIC obteve melhores resultados do que o método SIFT. As Figuras 81 e 82 ilustram os resultados obtidos.

Figura 80 – Pontos detectados pelo método SIFT - Exemplo 2.



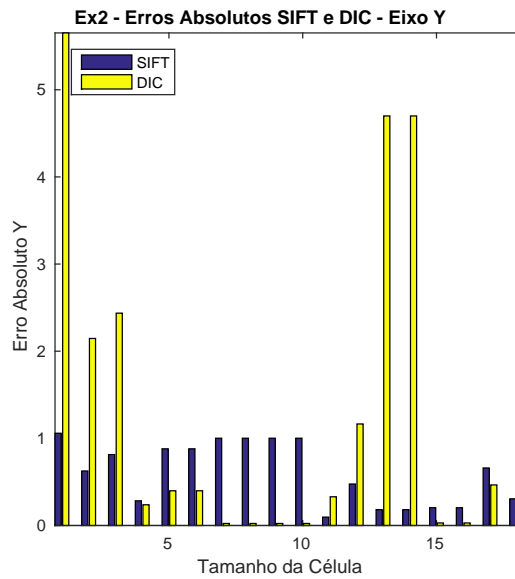
Fonte: Autor.

Figura 81 – Erros absolutos entre SIFT e DIC para o eixo X - Exemplo 2.



Fonte: Autor.

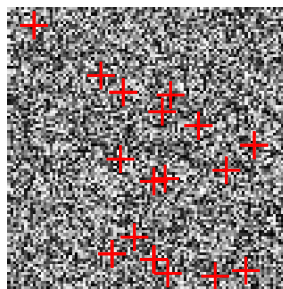
Figura 82 – Erros absolutos entre SIFT e DIC para o eixo Y - Exemplo 2.



Fonte: Autor.

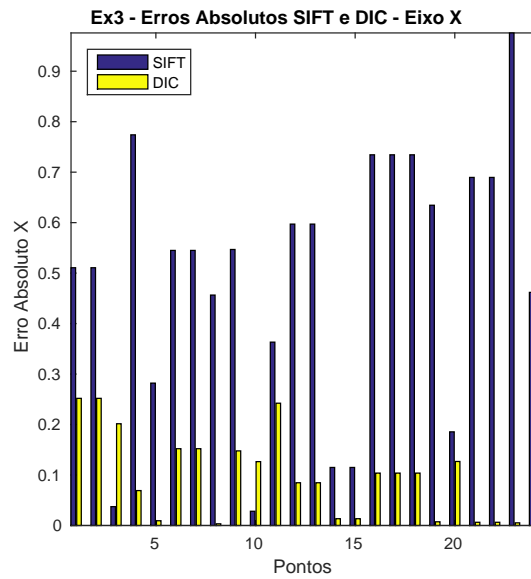
Para o terceiro exemplo, a Figura 83 mostra os pontos escolhidos pelo método SIFT. As Figuras 84 e 85 mostram os erros absolutos para os pontos escolhidos. Percebe-se que o método DIC apresenta valores de erros menores que aqueles apresentados pelo método SIFT, apesar de os valores serem relativamente baixos (o erro máximo encontrado foi 1 *pixel*, aproximadamente).

Figura 83 – Pontos detectados pelo método SIFT - Exemplo 3.



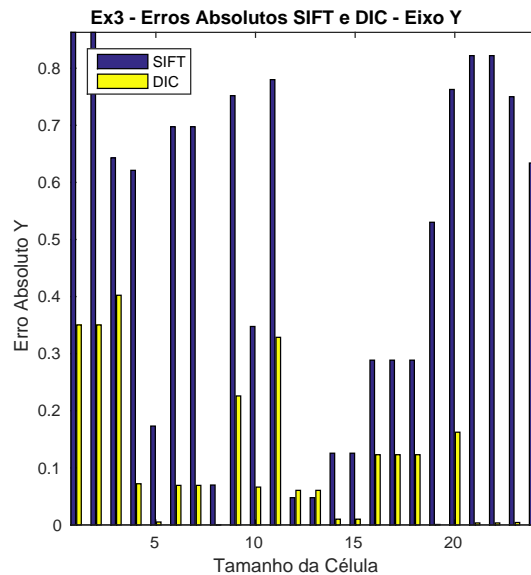
Fonte: Autor.

Figura 84 – Erros absolutos entre SIFT e DIC para o eixo X - Exemplo 3.



Fonte: Autor.

Figura 85 – Erros absolutos entre SIFT e DIC para o eixo Y - Exemplo 3.

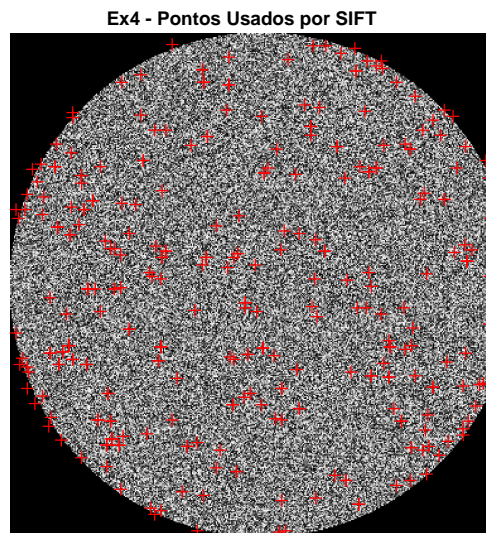


Fonte: Autor.

No quarto exemplo, o tamanho da imagem muda e, com isso, a quantidade de pontos capturadas pelo SIFT. A Figura 86 mostra os pontos captados pelo método. Os valores dos erros para os dois métodos foram semelhantes em todos os pontos analisados. Nota-se que há pontos onde o método DIC apresentou melhores resultados, assim como há locais onde o método SIFT foi melhor. Essa alteração ocorre em pontos de fronteira, onde o método SIFT apresenta melho-

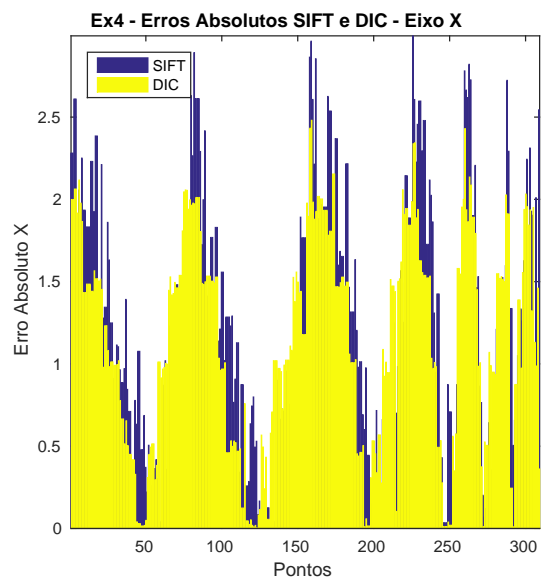
res resultados pois, como o método DIC necessita de informações da vizinhança, um conjunto de pontos com mesmo valor dificulta o cálculo de correlação.

Figura 86 – Pontos detectados pelo método SIFT - Exemplo 4.



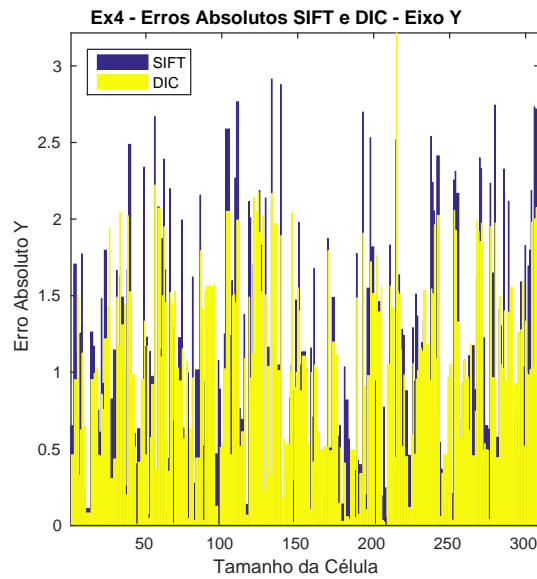
Fonte: Autor.

Figura 87 – Erros absolutos entre SIFT e DIC para o eixo X - Exemplo 4.



Fonte: Autor.

Figura 88 – Erros absolutos entre SIFT e DIC para o eixo Y - Exemplo 4.

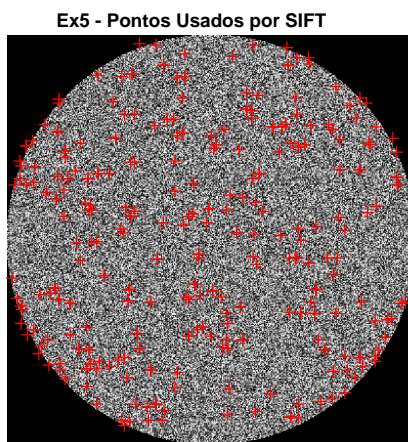


Fonte: Autor.

No quinto exemplo, cujo campo de deslocamentos é o mais elaborado, os valores absolutos dos erros foram maiores do que os erros apresentados nas aplicações anteriores. O método DIC apresenta melhores resultados quando os deslocamentos gerados são relativamente pequenos (da ordem 1 *pixel*).

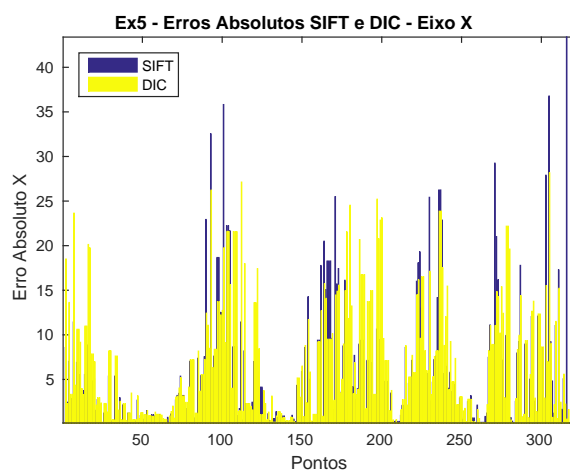
Assim, como o modelo de transformação usado para a deformação dessa imagem apresenta valores maiores que 1 *pixel*, principalmente na regiões próximas ao local de aplicação da carga pontual, o método não foi capaz de encontrar o valor correto. A Figura 89 mostra os pontos usados por SIFT. As Figuras 90 e 91 apresentam os valores dos erros absolutos para esse exemplo.

Figura 89 – Pontos detectados pelo método SIFT - Exemplo 5.



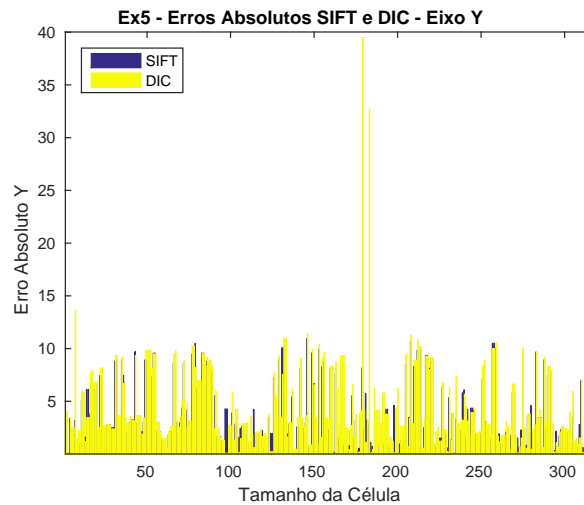
Fonte: Autor.

Figura 90 – Erros absolutos entre SIFT e DIC para o eixo X - Exemplo 5.



Fonte: Autor.

Figura 91 – Erros absolutos entre SIFT e DIC para o eixo Y - Exemplo 5.



Fonte: Autor.

De uma maneira geral, pode-se perceber que o método SIFT apresenta melhores resultados em alguns dos exemplos submetidos, quando comparado com o método DIC. É importante ressaltar que a localização irregular dos pontos escolhidos pelo método SIFT pode dificultar a otimização dos parâmetros do campo de deslocamentos a ser calculada pelo DIC.

O método SIFT não permite que sejam escolhidos pontos de interesse para cálculo pontual do deslocamento, uma vez que ele escolhe os melhores pontos seguindo os critérios do próprio método. Dessa forma, a irregularidade dos pontos dificulta a interpolação dos deslocamentos para determinação do comportamento dos deslocamentos na imagem.

5.5 Q4-DIC

5.5.1 Procedimento de Análise

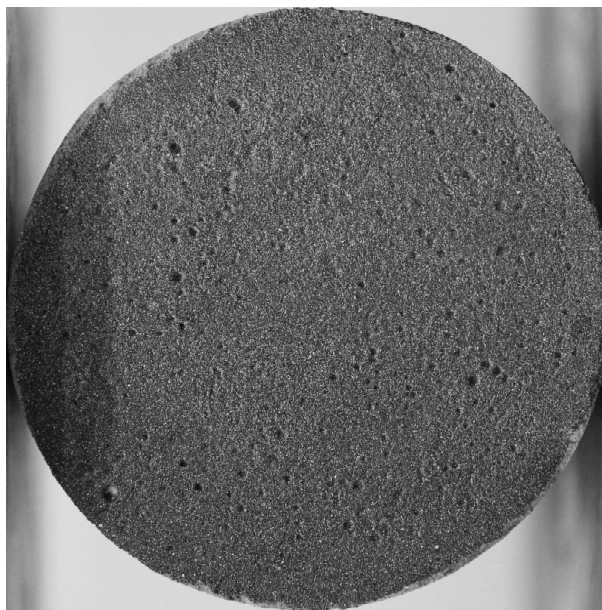
Sendo uma técnica de abordagem global, a análise realizada pelo Q4-DIC verifica apenas os pontos pertencentes à malha de pontos definida. Sua principal característica consiste na aplicação de funções de forma de elementos finitos (nesse caso, a função de forma do elemento linear Q4) para mapear os deslocamentos e garantir a continuidade do campo analisado em toda região de interesse. O ambiente computacional usado para análise foi o CorreliQ4, de Hild e Roux (2008).

Para esse exemplo, e diferente dos exemplos anteriores, foram usadas imagens reais de um ensaio de compressão diametral realizado e apresentado em Sombra e Codes (2015). No ensaio realizado por Sombra e Codes (2015), foram retiradas 41 imagens de 3321x3315 *pixels*, nas quais foi definida uma malha de 220 x 221 pontos, perfazendo um total de 48.620 pontos de análise. A Figuras 92 apresenta uma das imagens usadas.

Tabela 11 – Valores para os erros entre o protótipo e CorreliQ4.

	Mínimo (pixel)	Médio (pixel)	Máximo (pixel)	Desvio Padrão (pixel)
Eixo X	0	0,0947	1,1774	0,0788
Eixo Y	0	0,1023	1,1908	0,0877

Figura 92 – Imagem real do ensaio de compressão diametral.



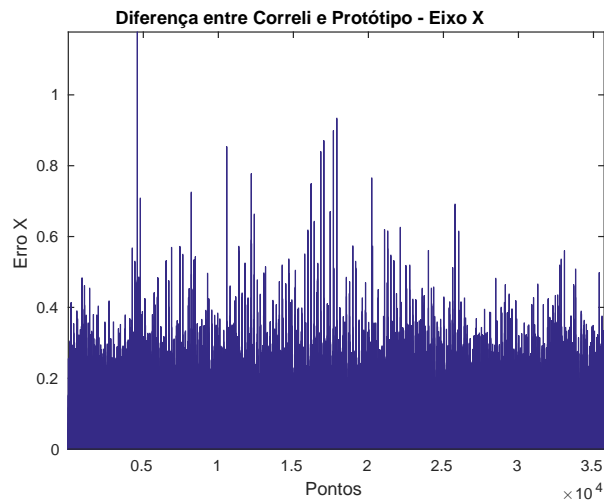
Fonte: Sombra e Codes (2015).

A análise usou as duas primeiras imagens do ensaio para realizar as comparações entre os pontos. O arquivo de resultados liberado por Sombra e Codes (2015) continha as informações de deslocamentos e coordenadas iniciais dos pontos usados para análise. Montou-se uma função para acessar o protótipo usando como parâmetros de entrada as coordenadas de cada um dos pontos. Por fim, compararam-se os resultados dos deslocamentos calculados pelo protótipo com os deslocamentos apresentados pelo CorreliQ4.

5.5.2 Resultados

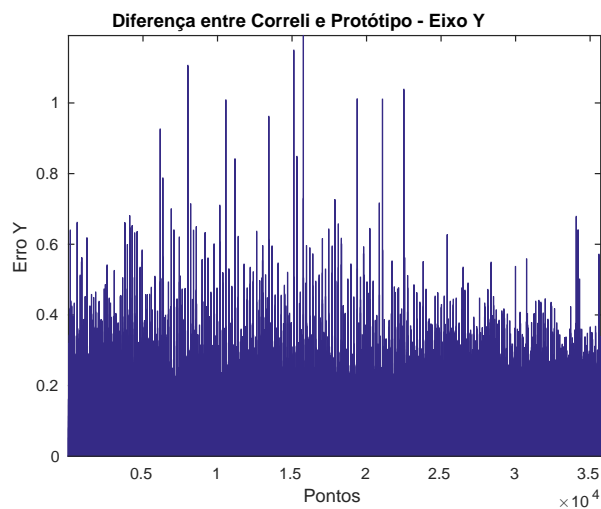
Conforme apresentado na seção anterior, a malha de pontos gerou 48.620 pontos. Como nem todos participavam do corpo de prova, uma parte dos pontos foi retirada. A análise trabalhou com aproximadamente 35.000 pontos, com resultados apresentados nas Figuras 93 e 94. A Tabela 11 apresenta os valores extremos para os gráficos apresentados.

Figura 93 – Erro entre Q4-DIC e protótipo - Eixo X.



Fonte: Autor.

Figura 94 – Erro entre Q4-DIC e protótipo - Eixo Y.



Fonte: Autor

Percebe-se, pelos resultados apresentados, que o protótipo apresentou valores muito próximos daqueles calculados pelo CorreliQ4. Em termos geométricos, a medida do diâmetro do corpo de prova analisado é de 10 cm, conforme Sombra e Codes (2015). Dividindo-se esse valor pela quantidade de *pixels* no eixo Y, encontra-se como escala 1 *pixel* equivalendo a 30 micrometros. Portanto, um erro médio de 0,1 *pixel* corresponde a um erro médio de 3 micrometros.

6 CONCLUSÃO

O rastreamento de pontos específicos de um elemento permite o cálculo dos deslocamentos internos do elemento fotografado, de maneira que informações com o campo de deformações, tensões e parâmetros constitutivos do material possam ser calculados. O presente trabalho propôs uma forma de obter os deslocamentos de pontos através de técnicas de análise e processamento de imagens. A Correlação de Imagens Digitais foi uma das técnicas de análises de imagens usada para correspondência de pontos.

Para o trabalho, foi planejada uma arquitetura computacional e um protótipo de um ambiente computacional foi implementado se baseando na arquitetura desenvolvida. Técnicas de Correlação de Imagens, otimização não-linear, interpolação bidimensional e Computação de Alto Desempenho foram incorporadas ao protótipo. A seguir, tratou-se da validação dos resultados obtidos pelo protótipo.

A validação do ambiente foi feita usando modelos de transformação de imagens baseados em campos de deslocamentos encontrados na literatura de Engenharia Estrutural. Os exemplos apresentavam diferentes graus de complexidade, de forma fosse possível fazer uma relação entre os resultados do ambiente e a complexidade do modelo de transformação.

Nesse caso, o ambiente computacional apresentou bons resultados no cálculo dos campos de deslocamentos até do segundo grau. O exemplo 3, de flexão pura, apresentou os melhores resultados entre os 3 primeiros exemplos. Observou-se também que o algoritmo apresenta melhores resultados quando o deslocamento dos *pixels* correspondentes é pequeno. Para os exemplos 4 e 5, que apresentam um modelo de transformação mais complexo, a correspondência foi inferior àquela obtida nos exemplos 1 a 3. Em especial, no exemplo 5 (de compressão diametral), foram encontrados os maiores erros. Nesse caso, observou-se que a própria geração da imagem constituiu um problema. O fato de haver uma singularidade no exemplo gerou deslocamentos com maiores erros na região mais próxima do ponto de aplicação da carga.

Também, o algoritmo de geração das imagens constituiu uma fonte de erro em todas as imagens geradas. A interpolação realizada pelo algoritmo, apesar de garantir continuidade C^1 à superfície plotada, inseriu um ruído de turvamento em todas as imagens. No ambiente real, esse borramento não é encontrado. Existe também a possibilidade de que o tamanho das células, muito pequenas quando comparadas com imagens reais (as imagens dos Exemplos 4 e 5 possuem 0,1 *megapixel*), tenha dificultado o cálculo da correlação em pontos mais distantes do centro da imagem. O uso de imagens geradas computacionalmente decorreu da falta de imagens reais de ensaios experimentais para um estudo prático. Mesmo na literatura acadêmica, nenhuma imagem em domínio público foi encontrada.

A implementação de técnicas de Computação de Alto Desempenho surgiu a partir da

análise do custo computacional das análises realizadas. Conforme apresentado na Seção 5, as análises mais longas tiveram custo computacional superior a 4000 segundos (≈ 67 minutos). A implementação de técnicas de Computação de Alto Desempenho permitiu que o tempo computacional fosse reduzido drasticamente: para um exemplo do mesmo tipo, a análise de 36 pontos foi realizada em aproximadamente 500 segundos. Em todos os exemplos realizados, o tempo computacional de análise foi reduzido e um *speed-up* máximo de 9x foi alcançado. O cálculo da escalabilidade também mostra que o aumento da carga de análise também leva ao aumento do *speed-up*.

A análise usando o método SIFT foi realizada para comparar a eficiência do método DIC em relação a outro método da área de Registro de Imagens. Em diversos pontos dos exemplos tratados, o SIFT foi capaz de apresentar menores erros absolutos que os erros calculados pelo método DIC. Nos exemplos 1 e 5, o método apresentou melhores resultados na maioria dos pontos. No entanto, para o Exemplo 5, os erros encontrados foram bastante altos, com erro máximo próximo aos 40 *pixels*. O SIFT, também apresentou melhores resultados para os pontos próximos à fronteira. Como o método DIC necessita das informações da vizinhança para o cálculo da correlação, pontos de fronteira nos Exemplos 4 e 5 mostraram erros elevados.

Em concordância com o uso do método SIFT, a aplicação do método Q4-DIC trouxe importantes resultados, uma vez que foi a primeira comparação realizada usando imagens reais, e campos de deslocamentos calculados por uma ferramenta de correlação de imagens bastante renomada no meio acadêmico. A comparação realizada apresentou resultados bastante satisfatórios, com uma variação média de 0.1 *pixel* de erro entre os valores calculados pelo protótipo e os valores calculados pelo CorreliQ4.

De uma maneira geral, o projeto mostrou que o cálculo de deslocamentos usando técnicas de análise e processamento de imagens constitui uma alternativa viável para análise de imagens de elementos estruturais. Há, no entanto, necessidade de implementação de um algoritmo robusto, capaz de filtrar ruídos nas imagens. Os dois métodos de análise apresentaram pontos onde cada um se destacou, o que indica que uma ação conjunta pode apresentar melhores resultados. Ainda assim, excelentes resultados foram obtidos, com o Exemplo 3 apresentando erros máximos da ordem do décimo do *pixel*, o que mostra o tipo de precisão que pode ser esperada do método DIC.

6.1 Trabalhos Futuros

O presente trabalho propôs uma arquitetura computacional para a detecção de campos de deslocamentos usando técnicas de processamento de imagens. O protótipo, desenvolvido usando uma técnica de abordagem local, foi capaz de encontrar com precisão a posição de determinado ponto em uma outra imagem.

Como sugestões de trabalhos futuros, dentro do aspecto do ambiente computacional, podem-se delimitar três vertentes:

1. O desenvolvimento do módulo de aquisição de imagens, com a implementação de modelos de arranjo e calibração de câmeras, geração de imagens digitais e aplicação de filtros para eliminação de ruídos. Os métodos para captação de imagens também podem ser usados para a construção de modelos computacionais 3D usando um arranjo com mais câmeras. A implementação inicial de filtros serviria para a redução de ruídos causados por mudanças de iluminação, problemas com a lente, entre outros.
2. Novas implementações no módulo de análise de imagens, com desenvolvimento de outras abordagens de correlação, indicadores de correlação, métodos de otimização não-linear, funções para mapeamento do campo de deslocamentos e deformações e interpolações bidimensionais. A implementação de métodos com abordagem global fornecem informações associadas à continuidade do campo de deslocamentos e deformações; o método espectral, conforme descrito por Chen et al. (1993), executa operações no domínio da frequência para calcular a correlação. Outras funções de mapeamento de campo, bem como outras interpolações e métodos de otimização não-linear podem ser usadas para acelerar o desempenho e melhorar a precisão para problemas específicos. Nesse módulo, há também a possibilidade de implementar técnicas diferentes de registro de imagens digitais, como a *Scale Invariant Feature Transform* (SIFT), conforme descrita por Rey Otero e Delbracio (2014).
3. Implementação de técnicas para visualização dos resultados, de forma a desenvolver o módulo de pós-processamento. Dentre algumas técnicas, citam-se as isofaixas, campos vetoriais e geração de vídeos. Um outro aspecto do módulo de pós-processamento envolve o desenvolvimento de técnicas estatísticas para avaliação de erros, com análises estocásticas de incertezas observadas no método, de forma a calcular um índice de confiabilidade do método usado ou do tamanho da região na qual o erro atinge determinado nível.

As sugestões anteriormente apresentadas são melhorias que podem ser desenvolvidas no protótipo ou na implementação de um ambiente computacional novo, usando uma linguagem de programação diferente. Outras sugestões envolvem uma redefinição do objetivo do ambiente computacional, com inclusão de novas funcionalidades.

Atualmente, o ambiente computacional visa apenas o cálculo de campos de deslocamentos. Esse tipo de informação pode ser usada para obtenção de outras grandezas físicas de Engenharia, como campos de tensões (conhecendo-se as características) do material. Há a possibilidade de caracterização de materiais, como coeficientes térmicos, pois os deslocamentos no material são provocados por fontes de calor. Na literatura, são encontrados métodos capazes de calcular também parâmetros constitutivos de materiais. Grédiac, Hild e Pineau (2012) apresenta uma revisão de alguns desses métodos.

Uma aplicação mais robusta do ambiente computacional envolve sua integração com ambientes de simulação numérica e ferramentas computacionais de projeto (*Computer Aided Design* - CAD). A integração visa comparação de valores entre modelos computacionais e seus correspondentes objetos físicos usados em ensaios experimentais. Outros tipos de integração envolvem análises multi-modais, onde vários campos de diferentes grandezas físicas são captados usando equipamentos com diversos tipos de sensores.

REFERÊNCIAS

- American Society for Testing and Materials. *Standard Guide for Evaluating Non-Contacting Optical Strain Measurement*. [S.l.: s.n.], 2013. v. 02. 1–7 p.
- ASSIS, W. S. *Sistemas computacionais de apoio à monitoração de estruturas de engenharia civil*. 296 p. Tese (Doutorado) — Universidade de São Paulo, 2007.
- AVRIL, S. et al. Overview of Identification Methods of Mechanical Parameters Based on Full-field Measurements. *Experimental Mechanics*, v. 48, n. 4, p. 381–402, aug 2008. ISSN 0014-4851.
- BANON, G. J. F. *Formal introduction to digital image processing*. São José dos Campos, SP, Brazil: INPE, 2000.
- BESNARD, G.; HILD, F.; ROUX, S. “Finite-Element” Displacement Fields Analysis from Digital Images: Application to Portevin–Le Châtelier Bands. *Experimental Mechanics*, v. 46, n. 6, p. 789–803, dec 2006. ISSN 0014-4851.
- BLABER, J.; ADAIR, B.; ANTONIOU, a. Ncorr: Open-Source 2D Digital Image Correlation Matlab Software. *Experimental Mechanics*, p. 1105–1122, 2015. ISSN 0014-4851.
- BONDI, A. Characteristics of Scalability and Their Impact on Performance. *Proceedings of the 2nd international workshop on Software and performance*, p. 195–203, 2000.
- BROWN, L. G. A survey of image registration techniques. *ACM Computing Surveys*, v. 24, n. 4, p. 325–376, dec 1992. ISSN 03600300.
- BRUCK, H. A. et al. Digital image correlation using Newton-Raphson method of partial differential correction. *Experimental Mechanics*, v. 29, n. 3, p. 261–267, 1989. ISSN 00144851.
- CHEN, D. J. et al. Digital speckle-displacement measurement using a complex spectrum method. *Applied Optics*, v. 32, n. 11, p. 1839, apr 1993. ISSN 0003-6935.
- CHEN, F. et al. Full-field 3D measurement using multi-camera digital image correlation system. *Optics and Lasers in Engineering*, Elsevier, v. 51, n. 9, p. 1044–1052, 2013. ISSN 01438166.
- CHENG, P. et al. Full-field Speckle Pattern Image Correlation with B-Spline Deformation Function. *Experimental Mechanics*, v. 42, n. September, p. 344–352, 2002. ISSN 00144851.
- CHU, T. C.; RANSON, W. F.; SUTTON, M. A. Applications of digital-image-correlation techniques to experimental mechanics. *Experimental Mechanics*, v. 25, n. 3, p. 232–244, sep 1985. ISSN 0014-4851.
- Correlated Solutions Inc. *VIC2D/3D*. Columbia: [s.n.], 2008.
- DUFOUR, J.-E. et al. CAD-based Displacement Measurements with Stereo-DIC. *Experimental Mechanics*, v. 55, n. 9, p. 1657–1668, 2015. ISSN 0014-4851.
- FERRAZ, D. H. S. *Obtenção de Campos de Deslocamentos e Deformações Através de Técnicas de Correlação de Imagens Digitais*. 114 p. Tese (Doutorado) — Universidade Federal de Alagoas, 2013.

- GATTASS, M. Método de Levenberg-Marquardt. *Algoritmo de Levenberg-Marquardt*, p. 10, 2008.
- GRÉDIAC, M. The use of full-field measurement methods in composite material characterization: interest and limitations. *Composites Part A: Applied Science and Manufacturing*, v. 35, n. 7-8, p. 751–761, 2004. ISSN 1359835X.
- GRÉDIAC, M.; HILD, F.; PINEAU, A. *Full-Field Measurements and Identification in Solid Mechanics*. 1. ed. [S.l.]: Wiley, 2012. 485 p. ISBN 9781848212947.
- HARVENT, J. et al. Shape Measurement Using a New Multi-step Stereo-DIC Algorithm That Preserves Sharp Edges. *Experimental Mechanics*, v. 55, n. 1, p. 167–176, 2014. ISSN 0014-4851.
- HILD, F.; ROUX, S. Correli Q4: A software for-finite-element-displacement field measurements by digital image correlation. *Internal report*, 2008.
- JIANG, Z. et al. Path-independent digital image correlation with high accuracy, speed and robustness. *Optics and Lasers in Engineering*, Elsevier, v. 65, p. 93–102, 2014. ISSN 01438166.
- Joint Committee For Guides In Metrology (JCGM). International vocabulary of metrology — Basic and general concepts and associated terms (VIM). *VIM3: International Vocabulary of Metrology*, v. 3, n. Vim, p. 104, 2008. ISSN 02632241.
- LECLERC, H. et al. Integrated Digital Image Correlation for the Identification of Mechanical Properties. *Computer Vision/Computer Graphics Collaboration Techniques*, p. 161–171, 2009.
- LOWE, D. Object recognition from local scale-invariant features. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. [S.l.]: IEEE, 1999. v. 2, n. [8, p. 1150–1157 vol.2. ISBN 0-7695-0164-8. ISSN 0-7695-0164-8.
- LOWE, D. G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, v. 60, n. 2, p. 91–110, nov 2004. ISSN 0920-5691.
- LU, H.; CARY, P. D. Deformation measurements by digital image correlation: Implementation of a second-order displacement gradient. *Experimental Mechanics*, v. 40, n. 4, p. 393–400, dec 2000. ISSN 0014-4851.
- LUCAS, B. D.; KANADE, T. An Iterative Image Registration Technique with an Application to Stereo Vision. *Imaging*, v. 130, n. x, p. 674–679, 1981. ISSN 17486815.
- MA, C. C.; HUNG, K. M. Exact full-field analysis of strain and displacement for circular disks subjected to partially distributed compressions. *International Journal of Mechanical Sciences*, v. 50, n. 2, p. 275–292, 2008. ISSN 00207403.
- MADSEN, K.; NIELSEN, H. B.; TINGLEFF, O. *Methods for Non-Linear Least Squares Problems*. 2. ed. Copenhagen: Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2004. 60 p.
- PAN, B. Recent Progress in Digital Image Correlation. *Experimental Mechanics*, v. 51, n. 7, p. 1223–1235, 2011. ISSN 00144851.

- PAN, B. et al. Digital image correlation using iterative least squares and pointwise least squares for displacement field and strain field measurements. *Optics and Lasers in Engineering*, v. 47, n. 7-8, p. 865–874, jul 2009. ISSN 01438166.
- PAN, B. et al. Comparison of Subset-Based Local and Finite Element-Based Global Digital Image Correlation. *Experimental Mechanics*, p. 887–901, 2015. ISSN 0014-4851.
- PETERS, W. H. et al. Application of digital correlation methods to rigid body mechanics. *Optical Engineering*, v. 22, n. 6, p. 738–742, 1983.
- PRESSMAN, R. S. *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman*. [S.l.: s.n.], 2009. 0 p. ISBN 978-0-07-337597-7.
- Rey Otero, I.; DELBRACIO, M. Anatomy of the SIFT Method. *Image Processing On Line*, v. 4, p. 370–396, dec 2014. ISSN 2105-1232.
- SCHREIER, H. W.; BRAASCH, J. R.; SUTTON, M. A. *Systematic errors in digital image correlation caused by intensity interpolation*. 2000. 2915 p.
- SCHREIER, H. W.; ORTEU, J.-J.; SUTTON, M. A. *Image correlation for shape, motion and deformation measurements: Basic concepts, theory and applications*. [S.l.: s.n.], 2009. 1–321 p. ISBN 9780387787466.
- SCHREIER, H. W.; SUTTON, M. A. Systematic errors in digital image correlation due to undermatched subset shape functions. *Experimental Mechanics*, v. 42, n. 3, p. 303–310, sep 2002. ISSN 0014-4851.
- SHEN, B.; PAULINO, G. H. Direct Extraction of Cohesive Fracture Properties from Digital Image Correlation: A Hybrid Inverse Technique. *Experimental Mechanics*, v. 51, n. 2, p. 143–163, 2011. ISSN 00144851.
- SMITH, T. S.; BAY, B. K.; RASHID, M. M. Digital Volume Correlation Including Rotational Degrees of Freedom during Minimization. *Experimental Mechanics*, v. 42, n. 3, p. 272–278, 2002. ISSN 00144851.
- SOMBRA, L. M.; CODES, R. N. *Análise do comportamento mecânico do concreto com agregados miúdos reciclados de resíduos de construção e demolição através do ensaio de compressão diametral com o uso da técnica de correlação de imagens*. 73 p. — UFERSA, 2015.
- SUN, Y. et al. Finite element formulation for a digital image correlation method. *Applied optics*, v. 44, n. 34, p. 7357–7363, 2005. ISSN 0003-6935.
- SUTTON, M. A.; HILD, F. Recent Advances and Perspectives in Digital Image Correlation. *Experimental Mechanics*, n. February, p. 1–8, 2015. ISSN 0014-4851.
- SUTTON, M. A. et al. Application of an optimized digital correlation method to planar deformation analysis. *Image and Vision Computing*, v. 4, n. 3, p. 143–150, aug 1986. ISSN 02628856.
- SUTTON, M. A. et al. Determination of displacements using an improved digital correlation method. *Image and Vision Computing*, v. 1, n. 3, p. 133–139, aug 1983. ISSN 02628856.

SUTTON, M. A. et al. The effect of out-of-plane motion on 2D and 3D digital image correlation measurements. *Optics and Lasers in Engineering*, v. 46, n. 10, p. 746–757, 2008. ISSN 01438166.

The MathWorks Inc. *Matlab 2015a*. 2015.

VENDROUX, G.; KNAUSS, W. G. Submicron deformation field measurements: Part 2. Improved digital image correlation. *Experimental Mechanics*, v. 38, n. 2, p. 86–92, 1998. ISSN 0014-4851.

WANG, B.; PAN, B. Random Errors in Digital Image Correlation Due to Matched or Overmatched Shape Functions. *Experimental mechanics*, p. 1–11, 2015. ISSN 0014-4851.

Xian-He S.; ROVER, D. Scalability of parallel algorithm-machine combinations. *IEEE Transactions on Parallel and Distributed Systems*, v. 5, n. 6, p. 599–613, jun 1994. ISSN 10459219.

ZITOVÁ, B.; FLUSSER, J. Image registration methods: A survey. *Image and Vision Computing*, v. 21, n. 11, p. 977–1000, 2003. ISSN 02628856.